

24. 1. 2005

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 4 年 8 月 1 9 日
Date of Application:

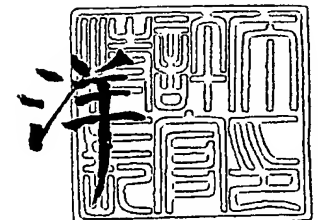
出 願 番 号 特 願 2 0 0 4 - 2 3 9 7 9 2
Application Number:
[ST. 10/C] : [J P 2 0 0 4 - 2 3 9 7 9 2]

出 願 人 キヤノン株式会社
Applicant(s):

2 0 0 5 年 3 月 1 7 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



BEST AVAILABLE COPY

【書類名】 特許願
【整理番号】 0002623-01
【提出日】 平成16年 8月19日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 7/00
【発明者】
 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社内
 【氏名】 御手洗 裕輔
【発明者】
 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社内
 【氏名】 真継 優和
【発明者】
 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社内
 【氏名】 森 克彦
【発明者】
 【住所又は居所】 広島県東広島市西条下見 5 丁目 1 0 - 7 株式会社エイアールテック内
 【氏名】 森江 隆
【特許出願人】
 【識別番号】 000001007
 【氏名又は名称】 キヤノン株式会社
【代理人】
 【識別番号】 100076428
 【弁理士】
 【氏名又は名称】 大塚 康德
【選任した代理人】
 【識別番号】 100112508
 【弁理士】
 【氏名又は名称】 高柳 司郎
【選任した代理人】
 【識別番号】 100115071
 【弁理士】
 【氏名又は名称】 大塚 康弘
 【連絡先】 担当
【選任した代理人】
 【識別番号】 100116894
 【弁理士】
 【氏名又は名称】 木村 秀二
【先の出願に基づく優先権主張】
 【出願番号】 特願2004- 24485
 【出願日】 平成16年 1月30日
【手数料の表示】
 【予納台帳番号】 003458
 【納付金額】 16,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 0102485

【書類名】特許請求の範囲

【請求項 1】

n 個のデータで構成される被符号化データを、m 個のデータで構成される適応化基底に符号化する方法であって、

前記適応化基底を初期化するステップと、

所定の変換により前記適応化基底から、n 個のデータで構成される復元データを生成する復元ステップと、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記適応化基底を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 2】

n 個のデータで構成される被符号化データを、少なくとも 2 つの、p 個のデータで構成される適応化基底に符号化する方法であって、

前記適応化基底を初期化するステップと、

所定の変換により前記少なくとも 2 つの適応化基底から、n 個のデータで構成される復元データを生成する復元ステップと、

所定の誤差評価関数により、前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記少なくとも 2 つの適応化基底を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 3】

n 個のデータで構成される被符号化データを、少なくとも 2 つの、p 個のデータで構成される適応化基底に符号化する方法であって、

前記適応化基底を初期化するステップと、

所定の変換により 1 つの前記適応化基底から、n 個のデータからなる要素復元データを生成する復元ステップと、

所定の誤差評価関数により、差分被符号化データと前記要素復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記 1 つの適応化基底を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 4】

被符号化データである k 次元空間分布データ $f(x_1, x_2, \dots, x_k)$ を、請求項 3 に記載の前記被符号化データを用いて、少なくとも 1 組の、k 個の 1 次元適応化基底組 $X_1(x_1)$ 、 $X_2(x_2)$ 、 \dots $X_k(x_k)$ を生成し、

$$f(x_1, x_2, \dots, x_k) = \sum_i \prod_k X_{ki}(x_k)$$

と展開表現することを特徴とするデータ符号化方法。

【請求項 5】

被符号化データである k 次元空間分布データ $f(x_1, x_2, \dots, x_k)$ を、k 個の 1 次元適応化基底組 $X_1(x_1)$ 、 $X_2(x_2)$ 、 \dots 、 $X_k(x_k)$ に符号化する方法であって、

前記 1 次元適応化基底組を初期化するステップと、

前記 1 次元適応化基底組から、復元データを次式で生成する復元ステップと、

$$\tilde{f}(x_1, x_2, \dots, x_k) = \prod_k X_k(x_k)$$

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記 1 次元適応化基底組を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 6】

被符号化データである k 次元空間分布データ $f(x_1, x_2, \dots, x_k)$ を、少なくとも 2 組の、 k 個の 1 次元適応化基底組 に符号化する方法であって、

前記 1 次元適応化基底組を初期化するステップと、

前記 1 次元適応化基底組から、復元データを次式で復元する復元ステップと、

$$\tilde{f}(x_1, x_2, \dots, x_k) = \sum \prod_k X_k(x_k)$$

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記 1 次元適応化基底組を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 7】

被符号化データである k 次元空間分布データを、少なくとも 2 組の、 k 個の 1 次元適応化基底組 に符号化する方法であって、

前記 1 次元適応化基底組を初期化するステップと、

1 組の前記 1 次元適応化基底組から、要素復元データを次式で生成する復元ステップと

$$\tilde{f}_i(x_1, x_2, \dots, x_k) = \prod_k X_k(x_k)$$

所定の誤差評価関数により、差分被符号化データと前記要素復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記 1 組の 1 次元適応化基底組を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 8】

前記復元ステップは、前記少なくとも 2 つの適応化基底の各々について、所定の変換で n 個のデータで構成される要素復元データに変換し、前記変換された要素復元データの線形和をとり、前記復元データとすることを特徴とする請求項 2 に記載のデータ符号化方法。

【請求項 9】

前記算出した誤差の収束を判定するステップを更に有し、前記誤差が収束したと判定されるまで、前記修正ステップを繰り返し行うことを特徴とする請求項 1、2、5、または 6 のいずれか 1 つに記載のデータ符号化方法。

【請求項 10】

前記算出した誤差の収束を判定するステップを更に有し、

前記誤差が収束したと判定されるまで、前記修正ステップを繰り返し行い、前記誤差が収束したと判定された場合に、次の新たな適応化基底、若しくは新たな 1 次元適応化基底組に対し、前記修正ステップを行う

ことを特徴とする請求項 3、または請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 11】

前記差分被符号化データは、前記被符号化データから、前記誤差の収束を判定するステップにより、前記誤差が収束したと判定された全ての前記適応化基底、若しくは前記 1 次元適応化基底組から復元される前記要素復元データを差し引いたものであることを特徴とする請求項 10 に記載のデータ符号化方法。

【請求項 12】

前記修正ステップを、前記算出した誤差が予め定めた目標誤差以下になるまで繰り返し行うことを特徴とする請求項 1 乃至請求項 3、請求項 5 乃至請求項 7 のいずれか 1 項に記

載の記載のデータ符号化方法。

【請求項 1 3】

前記誤差評価関数は、前記符号化データ、若しくは前記差分被符号化データと、前記復元データ、若しくは前記要素復元データの各要素との 2 乗誤差の総和であることを特徴とする請求項 1 乃至請求項 3、または請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 1 4】

前記修正ステップにおいて、前記算出した誤差が減少するように、前記適応化基底、若しくは前記 1 次元適応化基底組の修正を行うことを特徴とする請求項 1 乃至請求項 3、または請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 1 5】

前記修正ステップにおいて、前記適応化基底、若しくは前記 1 次元適応化基底組の修正量は、前記誤差評価関数の、前記適応化基底の内の修正する要素、若しくは前記 1 次元適応化基底組の内の修正する要素での偏微分係数に、所定の係数を乗じたものであることを特徴とする請求項 1 乃至請求項 3、または請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 1 6】

前記所定の係数は、負の定数であることを特徴とする請求項 1 5 に記載のデータ符号化方法。

【請求項 1 7】

前記所定の係数は、負の値であり、徐々に 0 に近づくことを特徴とする請求項 1 5 に記載のデータ符号化方法。

【請求項 1 8】

前記被符号化データを、所定範囲で分割し、前記所定範囲ごとに符号化を行うことを特徴とする請求項 1 乃至請求項 3、または請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 1 9】

前記被符号化データを、空間的に所定範囲で分割し、前記所定範囲ごとに符号化を行うことを特徴とする請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 2 0】

前記被符号化データを、周波数空間に変換し符号化を行うことを特徴とする請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 2 1】

前記被符号化データを、周波数空間に変換し、周波数空間において所定範囲に分割し、前記所定範囲ごとに符号化を行うことを特徴とする請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 2 2】

前記被符号化データを、空間的に所定範囲で分割し、前記所定範囲ごとに周波数空間に変換し符号化を行うことを特徴とする請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 2 3】

前記被符号化データを、空間的に所定範囲で分割し、前記所定範囲ごとに周波数空間に変換し、前記周波数空間に変換された所定範囲ごとの所定周波数成分を周波数データとして合成し、前記周波数データの符号化を行うことを特徴とする請求項 5 乃至請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 2 4】

請求項 1、5、2 3 のいずれか 1 項に記載のデータ符号化方法により生成される前記適応化基底、若しくは前記 1 次元適応化基底組から、前記所定の変換により復元データを復元することで、復号化することを特徴とするデータ復号化方法。

【請求項 2 5】

請求項 2、3、6、7、23 のいずれか 1 項に記載のデータ符号化方法により生成される前記適応化基底、若しくは前記 1 次元適応化基底組から、前記所定の変換により要素復元データを復元し、前記復元した要素復元データの線形和をとることで、復号化することとを特徴とするデータ復号化方法。

【請求項 26】

k 次元荷重分布データ $w(x_1, x_2, \dots, x_k)$ と、k 次元分布データである被積和演算データ $A(x_1, x_2, \dots, x_k)$ とを次式で積和演算を行う積和演算方法であって、

$$y = \iint \dots \int A(x_1, x_2, \dots, x_k) \cdot w(x_1, x_2, \dots, x_k) dx_1 dx_2 \dots dx_k$$

前記 k 次元荷重分布データを、請求項 5 乃至 7 のいずれか 1 項に記載のデータ符号化方法により、少なくとも 1 組の、k 個の 1 次元適応化基底組 $X_{11}(x_1)$ 、 $X_{21}(x_2)$ 、 \dots 、 $X_{k1}(x_k)$ に符号化し、

前記被積和演算データと、前記 1 次元適応化基底組を用いて、前記積和演算を次式で求める

$$y = \sum_i \left[\iint \dots \int A(x_1, x_2, \dots, x_k) \cdot \prod_i X_{i1}(x_i) dx_1 dx_2 \dots dx_k \right]$$

ことを特徴とする積和演算方法。

【請求項 27】

n 個のデータで構成される被符号化データを、m 個のデータで構成される適応化基底に符号化する装置であって、

前記適応化基底を初期化する手段と、

所定の変換により前記適応化基底から、n 個のデータからなる復元データを生成する復元手段と、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出手段と、

前記誤差に基づき前記適応化基底を修正する修正手段を有することを特徴とするデータ符号化装置。

【請求項 28】

k 次元の被符号化データを、k 個の 1 次元適応化基底の組に符号化するデータ符号化方法であって、

被符号化データを所定のメモリに記憶する記憶ステップと、

前記 k 個の 1 次元適応化基底をメモリに確保し、初期化する初期化ステップと、

所定の変換により、前記メモリ内の 1 次元適応化基底から、k 次元の復元データを生成する復元ステップと、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

該誤差演算ステップで得られた誤差に基づき、前記適応化基底を修正するか否かを判定し、前記適応化基底を修正すると判定した場合には前記メモリの適応化基底を修正して、前記復元ステップ、誤差演算ステップを繰り返し、前記適応化基底を修正しないと判定した場合には、最後の適応化基底を符号化結果として出力する判定ステップと

を備えることを特徴とするデータ符号化方法。

【請求項 29】

前記判定ステップは、誤差が収束したか否かに基づいて適応化基底を修正するか否かを判定し、

収束したと判定した場合であって、被符号化データと前記復元データとの誤差が所定値以上の差を有する場合には、k 個の 1 次元適応化基底の組を更に追加することを特徴とする請求項 28 に記載のデータ符号化方法。

【請求項 30】

k 次元の被符号化データを、k 個の 1 次元適応化基底の組に符号化するデータ符号化方

法であって、

被符号化データを所定のメモリに記憶する被符号化データ記憶手段と、
前記 k 個の一次元適応化基底を記憶する基底記憶手段と、
該基底記憶手段に記憶された k 個の一次元適応化基底を初期化する初期化手段と、
所定の変換により、前記基底記憶手段に記憶された k 個の一次元適応化基底から、k 次元の復元データを生成する復元手段と、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出手段と、

該誤差演算手段で得られた誤差に基づき、前記適応化基底を修正するか否かを判定し、前記適応化基底を修正すると判定した場合には前記基底記憶手段に記憶された適応化基底を修正して、前記復元手段、誤差演算手段を繰り返し、前記適応化基底を修正しないと判定した場合には、最後の適応化基底を符号化結果として出力する判定手段と、

を備えることを特徴とするデータ符号化装置。

【請求項 31】

k 次元荷重分布データ $w(x_1, x_2, \dots, x_k)$ と、k 次元分布データである被積和演算データ $A(x_1, x_2, \dots, x_k)$ とを次式で積和演算を行う積和演算装置であって、

$$y = \iint \dots \int A(x_1, x_2, \dots, x_k) \cdot w(x_1, x_2, \dots, x_k) dx_1 dx_2 \dots dx_k$$

前記 k 次元荷重分布データを、請求項 5 乃至 7 のいずれか 1 項に記載のデータ符号化方法により、少なくとも 1 組の、k 個の 1 次元適応化基底組 $X_{11}(x_1)$ 、 $X_{21}(x_2)$ 、 \dots 、 $X_{k1}(x_k)$ に符号化し、

前記被積和演算データと、前記 1 次元適応化基底組を用いて、前記積和演算を次式で求める

$$y = \sum \left[\iint \dots \int A(x_1, x_2, \dots, x_k) \cdot \prod_k X_{ki}(x_k) dx_1 dx_2 \dots dx_k \right]$$

ことを特徴とする積和演算装置。

【請求項 32】

前記積和演算装置において、積和演算を行なう演算素子がアレイ状に構成されていることを特徴とする請求項 31 記載の積和演算装置。

【請求項 33】

前記積和演算装置において、1 次元適応化基底組がアレイ状に構成された演算素子全てに入力されることを特徴とする請求項 32 に記載の積和演算装置。

【請求項 34】

前記積和演算装置において、アレイ状に構成された演算素子全てにおいて前記積和演算が実行されることを特徴とする請求項 33 に記載の積和演算装置。

【請求項 35】

前記積和演算装置において、1 次元適応化基底組が、アレイ状に構成された演算素子の列・行のそれぞれ一部に入力されることを特徴とする請求項 32 に記載の積和演算装置。

【請求項 36】

前記積和演算装置において、アレイ状に構成された演算素子の一部において前記積和演算が実行されることを特徴とする請求項 35 に記載の積和演算装置。

【書類名】明細書

【発明の名称】符号化方法及び装置、並びにコンピュータプログラム及びコンピュータ可読記憶媒体

【技術分野】

【0001】

本発明は、データの符号化、特に2次元画像や3次元密度分布等の多次元空間における分布データの符号化技術に関するものである。

【背景技術】

【0002】

2次元画像等の多次元分布データを符号化する技術としては、JEPG形式の圧縮技術のような、2次元画像データに対し離散コサイン変換を行い、変換されたデータに所定の量子化を行い、それをハフマン符号化等により符号化することで、データ量を圧縮する技術等が一般的である。

【0003】

また、画像データを予測符号化し、予測符号化による予測誤差データをブロック分割し、ブロック毎に予測誤差の大きいブロックか小さいブロックかを判別しする手法もある（特許文献1）。この文献では、予測誤差の大きいブロックにはベクトル量子化を適用し、ベクトル量子化により検索されたコードブックのベクトル値と予測誤差との差分値を計算し、この差分値と予測誤差の小さいブロックにおける予測誤差をエントロピー符号化し、エントロピー符号とブロック大小判別フラッグとベクトル量子化によるインデックスとを用いて符号化データを作成するものである。このように、省スペースでのデータ保持、高速通信等の観点から、高効率なデータ符号化技術が望まれている。

【0004】

また、まったく別の観点ではあるが、画像処理等で多用される所定の2次元荷重分布データによる、2次元画像に対する離散的コンボリューション演算を行う演算器において、積和演算素子を大規模並列化し、高速な演算を可能にすることが期待されている。しかし配線の問題等から、積和演算を2次元的に並列化して実行することは困難であった。従って、コンボリューションの2次元荷重分布データを1次元の基底組に分解する方法が望まれている。

【特許文献1】特開平11-331852号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

本発明は、このような点に鑑みなされたものであり、高効率なデータの符号化、特に2次元画像等の多次元空間における分布データ等を、幾つかの一次元の基底に分解し、符号化する技術を提供とするものである。

【課題を解決するための手段】

【0006】

この課題を解決するため、例えば本発明の符号化方法は以下の工程を備える。すなわち、

n個のデータで構成される被符号化データを、m個のデータで構成される適応化基底に符号化する方法であって、

前記適応化基底を初期化するステップと、

所定の変換により前記適応化基底から、n個のデータで構成される復元データを生成する復元ステップと、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記適応化基底を修正する修正ステップとを有する。

【発明の効果】

【0007】

本発明によれば、高効率なデータの符号化、特に2次元画像等の多次元空間における分布データ等を、高効率に符号化することができる。また2次元のデータを1次元の基底データに分解することが可能になるため、コンボリューション演算器において問題となっていた、2次元並列化による配線の問題を解決することができる。

【発明を実施するための最良の形態】

【0008】

以下、添付図面に従って本発明に係る実施形態を詳細に説明する。

【0009】

<第1の実施形態>

第1の実施形態として、有限範囲で定義される2次元濃淡画像を、2つの1次元適応化基底に符号化する方法を説明する。

【0010】

図8は本第1の実施形態における装置構成を示すブロック図である。

【0011】

図中、1は装置全体の制御を司るCPUであり、2はCPU1の処理手順（プログラム）を記憶しているROMである。3はCPU1のワークエリアとして使用されるRAMであり、ここには符号化対象の画像データを格納する画像データエリア3a、並びに、基底データを格納するエリア3bが確保される。なお、エリア3a、3bのサイズは入力する画像データサイズに応じて決定されることになる。4は符号化対象の濃淡画像データを入力する画像入力部である。画像入力部としては、イメージスキャナ、撮像素子等であるが、画像データが記憶媒体に記憶されている場合にはその記憶媒体をアクセスする装置であってもよいし、ネットワークを介して画像を受信する場合にはネットワークインタフェースであっても構わない。5は符号化後のデータを格納する外部記憶装置（ハードディスク等）であるが、符号化データをネットワーク上に出力する場合にはネットワークインタフェースでも構わない。

【0012】

実施形態では説明を簡単なものとするため、符号化対象の画像データは1画素8ビットの多値（濃淡）モノクロ画像として説明する。また、画像は矩形であり、水平方向の画素数が x_{max} 、垂直方向の画素数が y_{max} であるものとする。画像入力部4がイメージスキャナであれば、読取り解像度、原稿のサイズにより、 x_{max} 、 y_{max} は一義的に決まるものである。

【0013】

入力した画像は画像データエリア3aに格納されるが、座標 (x, y) （但し、 $0 \leq x < x_{max}$ 、 $0 \leq y < y_{max}$ ）の画素値を便宜的に $f(x, y)$ と表現する。また、入力する画像データのサイズが決定されるので、2つの1次元適応化基底 $X(x)$ 、 $Y(y)$ を基底エリア3bに確保する。 x 、 y の取り得る範囲は入力画像のそれと同じである。なお、画素値 $f(x, y)$ は8ビットの整数値であるが、1次元適応化基底 $X(x)$ 、 $Y(y)$ は小数点の値を記憶保持できるように多バイト形式でRAM3b内の基底エリア3bに確保することになる。

【0014】

以下、本第1の実施形態におけるCPU1の符号化処理手順を図1のフローチャート（対応するプログラムはROM2に格納されている）に従って説明する。

【0015】

まず、画像入力（ステップS10）において、符号化される2次元濃淡画像である画像 $f(x, y)$ を入力し、画像エリア3aに格納する。次に、基底設定を行う（ステップS11）。ここでは、2つの1次元適応化基底 $X(x)$ 、 $Y(y)$ について初期値を設定する。但し、この1次元適応化基底 $X(x)$ 、 $Y(y)$ は、 $0 \leq x < x_{max}$ 、 $0 \leq y < y_{max}$ という有限範囲全域において、全てが0ではないような値を設定する。初期段階では乱数関数を用いて設定することになる。なお、ここでは上記のように各定義域を有限としたが、特にこれに限るものではない。すなわち、無限の定義域において定義された $f(x, y)$

)、 $X(x)$ 、 $Y(y)$ においても、後述の誤差 E 、及び修正量 $\Delta X(x)$ 、 $\Delta Y(y)$ が発散しない限り、本手法は適用可能である(但し、メモリ容量は有限であるので、必ずその限界はある)。

【0016】

基底復元(ステップS12)では、復元画像 $f^-(x, y)$ を次のように生成する。

【数1】

$$\tilde{f}(x, y) = X(x) \cdot Y(y)$$

【0017】

次いで、ステップS13の誤差演算処理にて、入力画像 $f(x, y)$ と復元画像 $f^-(x, y)$ との誤差 E を、次の誤差評価関数を用いて算出する。

【数2】

$$E = \iint \frac{1}{2} \{f(x, y) - \tilde{f}(x, y)\}^2 dx dy$$

ここで、 $\int f(x) dx$ は、 x の定義領域全範囲における $f(x)$ の積分を示す。 y 成分についても同様に積分することになるので、誤差 E は入力画像全域における累積誤差となる。また、この誤差 E は、後述する誤差収束判定(分岐判定であるステップS16)において用いるため、履歴としてRAM3に順次記録しておく。

【0018】

続いて、この誤差 E に基づいて、修正量の算出を行う(ステップS14)。ここでは、各1次元適応化基底の修正量 $\Delta X(x)$ 、 $\Delta Y(y)$ を以下のように算出する。

【数3】

$$\Delta X(x) = -\eta_x \cdot \frac{\partial E}{\partial X(x)} = \eta_x \cdot \int \{f(x, y) - \tilde{f}(x, y)\} \cdot Y(y) dy$$

$$\Delta Y(y) = -\eta_y \cdot \frac{\partial E}{\partial Y(y)} = \eta_y \cdot \int \{f(x, y) - \tilde{f}(x, y)\} \cdot X(x) dx$$

ここで、 η_x 、 η_y は修正パラメータであり正の定数である(この η_x 、 η_y の設定については後述する)。

【0019】

この修正量を得る関数は、誤差 E を求める式の偏微分式であることに注意されたい。すなわち、誤差 E は常に0以上の値を有するわけであり、究極的には、本実施形態の目標は誤差 E が極小値となる1次元適応化基底を求める点にある。現時点での1次元適応化基底による傾きがマイナスであるということは、極小値となる一次元基底は現時点より正の方向に位置することになり、逆に、1次元適応化基底による傾きがプラスであるということは、極小値となる一次元基底は現時点より負の方向に存在することを意味する。それ故、傾きに対してマイナスを乗算することで、傾きに応じた修正量を算出することになる。

【0020】

次に、処理はステップS15に進み、基底修正処理を行う。これは、ステップS14にて得られた修正量 $\Delta X(x)$ 、 $\Delta Y(y)$ を修正前の1次元適応化基底を $X(x)$ 、 $Y(y)$ に加算することで、修正後の1次元適応化基底を $X'(x)$ 、 $Y'(y)$ を求めるものである。式で示せば次の通りである。

$$X'(x) = X(x) + \Delta X(x)$$

$$Y'(y) = Y(y) + \Delta Y(y)$$

この1次元適応化基底の修正の際には、入力画像 $f(x, y)$ が、定義された全ての x 、 y の範囲において常に0でない限り、各1次元適応化基底が常に0ではないという上記条件を満たさなければならない。ただし、そのような条件になってしまった場合は、1次元適応化基底の少なくとも一部にランダムに微小の揺らぎを与えるようにすればよい。

【0021】

上記のステップS12（基底復元処理）からステップS15（基底修正処理）までの1次元適応化基底の修正を、修正された1次元適応化基底に対して、誤差下限判定（ステップS16）、及び誤差収束判定（ステップS17）において終了判定が出されるまで繰り返し実行する。

【0022】

ここで、ステップS16の誤差下限判定は、誤差算出（ステップS13）で算出した直前の修正の誤差Eが、予め定めておいた所定の値以下であった場合に、その時の1次元適応化基底が許容範囲の誤差Eを与える1次元適応化基底であるか否かを判定するものである。そして、終了判定の場合には、その時点での1次元適応化基底を符号化データとして記録して処理を終了する。

【0023】

また、ステップS17の誤差収束判定は、誤差算出（ステップS13）で算出し、記録した誤差Eの履歴から、誤差の収束判定を行い、誤差が収束したと判定された場合は、その時の1次元適応化基底が誤差の極小値を与える1次元適応化基底であるとして終了判定する。収束したと判断した場合の1次元適応化基底は符号化データとして記録して処理を終了する。誤差の収束の判定としては、1回の修正ごとの、誤差Eの変化がほとんどない状況（誤差の差分が所定閾値以下の場合）が続いた場合に、誤差が収束したと判定する。

【0024】

上記説明した処理により、2次元濃淡画像 $f(x, y)$ を、2つの1次元適応化元基底 $X(x)$ 、 $Y(y)$ に符号化することができる。換言すれば、画像データ $f(x, y)$ は画素数は $x_{\max} \times y_{\max}$ 画素存在していたのに対し、符号化データは $x_{\max} + y_{\max}$ 個のデータにまで圧縮されたことになる（ただし、入力画像の1画素は8ビットであるのに対し、1次元適応化元基底の1要素は数バイト必要になる）。また、この符号化された2つの1次元適応化基底 $X(x)$ 、 $Y(y)$ の復号化画像 $f_{\text{decode}}(x, y)$ は、次のようにして復号できる。

$$f_{\text{decode}}(x, y) = X(x) \times Y(y)$$

上記符号化処理において、基底設定（ステップS11）において設定する初期の1次元適応化基底 $X(x)$ 、 $Y(y)$ としては、定義された x 、 y の範囲において $X(x)$ 、 $Y(y)$ は入力画像のレンジ程度の値であることが好ましい。好適には、0ではないものの、ほぼ0に近い値となるように設定すると、初期値依存性が少なくなり、適切な1次元適応化基底を得られる可能性が高くなる。

【0025】

また、修正パラメータ η_x 、 η_y は、入力画像 $f(x, y)$ のレンジや定義された x 、 y の範囲の幅に応じて定めるのが好ましい。具体的には η_x は y の範囲の2乗と $f(x, y)$ のレンジに反比例し、 η_y は x の範囲の2乗と $f(x, y)$ のレンジに反比例させるようにするのが好適である。ここでは修正パラメータを η_x 、 η_y と各1次元適応化基底により異ならせているが、定義された x 、 y の範囲の幅が同程度であれば、同一の値を用いても構わない。この修正パラメータを大きい値に設定すると、少ない回数の1次元適応化基底の修正で誤差下限判定（ステップS16）、若しくは誤差収束判定（ステップS17）での終了判定が出る可能性が高くなるが、誤差Eが極小値周辺で大きく振動し、適切な1次元適応化基底を得られない可能性が高くなる。また、小さい値に設定すると、誤差Eが極小値付近に落ち着いて、適切な1次元適応化基底を得られる可能性が高くなるが、誤差下限判定（ステップS16）、若しくは誤差収束判定（ステップS17）での終了判定までに、多くの回数の、1次元適応化基底の修正を必要とする可能性が高くなる。上記の点を鑑み、修正パラメータは、誤差Eが極小値周辺で大きく振動しない程度に小さく、極端に多くの回数の、1次元適応化基底の修正を必要としない程度の大きな値にしておくことが好適である。また修正パラメータを修正回数に応じて徐々に小さくしていく手法を適用しても良い。

【0026】

また、誤差下限判定（ステップS16）で用いる、予め定めておく誤差Eの許容範囲も、入力画像 $f(x, y)$ のレンジや、定義された x, y の範囲の幅に応じて定めるのが好ましく、これを大きい値にすると、少ない回数の1次元適応化基底の修正で誤差下限判定（ステップS16）での終了判定が出る可能性が高くなるが、誤差Eが高い1次元適応化基底を得る可能性が高くなる。また、小さい値にすると、誤差が小さい適切な1次元適応化基底を得られる可能性が高くなる。しかし、誤差下限判定（ステップS16）での終了判定までに、多くの回数の、1次元適応化基底の修正を必要とする、若しくは誤差下限判定（ステップS16）での終了判定が出ない可能性が高くなる。

【0027】

さらに、上記説明した修正量算出（ステップS14）で算出する修正量 $\Delta X(x)$ 、 $\Delta Y(y)$ を、予め保持しておいた、この修正の前の修正量 $\Delta \hat{X}(x)$ 、 $\Delta \hat{Y}(y)$ を用いて、以下のようにすることで、より誤差の小さい好適な1次元適応化基底を得られる、若しくは少ない修正回数で所望の1次元適応化基底を得られる可能性が高くなる。

【数4】

$$\Delta X(x) = \eta_x \cdot \int \{f(x, y) - \tilde{f}(x, y)\} \cdot Y(y) dy + \alpha_x \cdot \Delta \hat{X}(x)$$

$$\Delta Y(y) = \eta_y \cdot \int \{f(x, y) - \tilde{f}(x, y)\} \cdot X(x) dx + \alpha_y \cdot \Delta \hat{Y}(y)$$

ここで α_x 、 α_y は、慣性項係数であり、1未満の正の定数である。ここでは、慣性項係数を各1次元適応化基底により異ならせているが、同一の値を使っても構わない。ここで用いる慣性項は、大きい値に設定する程、上記利点が発生する可能性が高くなるが、上記利点と逆の効果も現れ、1次元適応化基底の修正が不安定になる可能性が高くなるため、あまり大きな値に設定することは好ましくない。また小さい値に設定すると、上記利点の逆の効果が現れる可能性は低くなるが、上記利点を得られる可能性は少なくなる。そのため、この慣性項係数は、1次元適応化基底の修正が不安定にならないような範囲でできる限り大きな値に設定することが好ましい。

【0028】

以上、第1の実施形態では、有限範囲で定義される2次元濃淡画像 $f(x, y)$ を、2つの1次元適応化基底 $X(x)$ 、 $Y(y)$ に符号化する方法の例として説明したが、符号化対象は2次元に限らない。例えば3次元の空間における密度分布等にも適用可能であるし、それを越える多次元に対しても同様に適用可能である。

【0029】

<第2の実施形態>

上記第1の実施形態では、符号化される2次元画像が比較的単純なものであれば、座標系を適切に選択することにより、誤差の少ない符号化が可能である。つまり、例えば2次元画像が、図5に示すような単純な矩形形状のものであれば、その矩形に直交する直交座標系を用いて符号化すれば、比較的誤差の少ない符号化が可能になる。このようにデータに応じて極座標系等や、3次元データであれば円柱座標系等を適切に選択してやることで、誤差Eが少ない符号化が可能である。

【0030】

しかし、2次元画像が例えば自然画像であるような場合は、上記第1の実施形態で示した方法では、符号化されたデータを用いて復号したデータ（画像データ）と、元のデータの誤差が大きくなってしまう。そこで、本第2の実施形態では、上記第1の実施形態の変形例として、有限範囲で定義される離散的な2次元濃淡画像 $f(x, y)$ を、複数の離散的な1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ (x, y は整数) に符号化する方法を説明する。

【0031】

概要を簡単に説明すると、1つの1次元適応化基底を用意し、第1の実施形態と同様の処理を行って誤差Eの下限判定、並びに、収束判定を行う。その後、誤差Eが十分に小さ

いと判断されない場合には、新たな一次元適応化基底組を追加し残された被符号化データと復元データの誤差を小さくするように、再度、同じ処理を繰り返すものである。

【0032】

装置構成は、第1の実施形態と同じとし、以下、本第2の実施形態に係る処理を図2のフローチャートに従って説明する。

【0033】

まず、画像入力（ステップS20）において、符号化される離散的な2次元濃淡画像である入力画像 $f(x, y)$ を入力する。ここでも第1の実施形態と同様に、 x, y は所定の定義域をもち、 $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$ とする。また、本第2の実施形態においては、入力画像は離散的であり、 x, y は整数である。

【0034】

ここでは、具体的な説明のため、例えば図10に示す 8×8 の輝度値データに対して当該処理を行った場合について説明していく。以下、この具体的な数値を用いた例を数値適用例と呼ぶ。図10に示した数値は、8bit精度の非負整数値であるが、差分が定義できる値であればどのようなデータでも構わない。画像入力（ステップS20）の次は、基底設定処理を行う（ステップS21）。ここでは、離散的な1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ (x, y は整数) を1つ選択して初期の1次元適応化基底組として設定する。添字「1」は基底番号であり、特に限定はしないが、本第2の実施形態では0から順に、予め定めた基底組数まで順に選択していくことにする。数値適用例で処理を行う画像は、 8×8 であるので、各1次元適応化基底は、8の要素からなる数値列となる。第1の実施形態で説明したように、ここでの各数値の初期設定は、ほぼ0と見なせる乱数を設定するのが好適であるが、全ての要素が0ではないという条件を満たせばよいので、ここでは簡単のため、全ての添字1の、初期の1次元適応化基底組において、図11に示すような1次元適応化基底組を用いる。図中の式の、括弧内の t の値は、1次元適応化基底の修正回数を示している。つまり、 $t = n$ ならば、 n 回の修正後の1次元適応化基底を意味する。

【0035】

次に基底復元処理を行う（ステップS22）。ここでは、選択した基底番号1の1次元適応化基底組から、要素復元画像 $f_1^-(x, y)$ を次のように算出する。

【数5】

$$\tilde{f}_1(x, y) = X_1(x) \cdot Y_1(y)$$

【0036】

基底設定処理（ステップS21）直後の、1番最初の基底組を用いた時の、要素復元画像 $f_0^-(x, y)$ に対応する 8×8 の数値列は図12に示すようになる。次に、誤差算出（ステップS23）により、誤差 E を以下のように算出する。

【数6】

$$E = \sum_x \sum_y \left[\frac{1}{2} \{ f_1(x, y) - \tilde{f}_1(x, y) \}^2 \right]$$

上式において、 $f_1(x)$ は差分被符号化画像であり、以下のように表される。

【数7】

$$f_1(x, y) = f(x, y) - \sum_{n=0}^{l-1} \tilde{f}_n(x, y)$$

ここで、

【数8】

$$\sum_x f(x)$$

は、 x の定義域内全範囲における、 x が整数の時の $f(x)$ の総和を意味する。

【0037】

また、

【数9】

$$\sum_{k=n}^m f_k(x)$$

は、 $n \leq k \leq m$ の $f_k(x)$ の線形和を意味し、 $m < n$ の場合は 0 となる。数値適用例では、差分被符号化画像 $f_0(x)$ は、入力画像となるので、誤差 E は 482913 である。

。

【0038】

上記の差分被符号化画像 $f_1(x, y)$ は、入力画像 $f(x, y)$ から、基底番号 0 から基底番号 1-1 までの、1 組の 1 次元適応化基底組により復元される各要素復元画像を差し引いたものを意味する。つまり、本第 2 の実施形態では基底番号 0 から順に 1 次元適応化基底組を選択していき、各 1 次元適応化基底組を修正していくので、差分被符号化画像 $f_1(x, y)$ は、入力画像 $f(x, y)$ から、既に修正完了した 1 次元適応化基底組により復元される各要素復元画像を差し引いたものとなる。

【0039】

次いで、修正量算出（ステップ S24）において、基底設定（ステップ S21）において選択された 1 次元適応化基底組 $X_1(x)$ 、及び $Y_1(y)$ に対する修正量 $\Delta X_1(x)$ 、及び $\Delta Y_1(y)$ を以下のように算出する。

【数10-A】

$$\Delta X_1(x) = -\eta_x \cdot \frac{\partial E}{\partial X_1(x)} = \eta_x \cdot \sum_y [f_1(x, y) - \tilde{f}_1(x, y)] \cdot Y_1(y)$$

$$\Delta Y_1(y) = -\eta_y \cdot \frac{\partial E}{\partial Y_1(y)} = \eta_y \cdot \sum_x [f_1(x, y) - \tilde{f}_1(x, y)] \cdot X_1(x)$$

若しくは、第 1 の実施形態で述べた、慣性項計数 α_x 、 α_y を前回の修正量 $\Delta \hat{X}(x)$ 、 $\Delta \hat{Y}(y)$ に乗じたものを加算した、以下の式を用いて算出しても構わない。

【数10-B】

$$\Delta X_1(x) = -\eta_x \cdot \frac{\partial E}{\partial X_1(x)} + \alpha_x \cdot \Delta \hat{X}(x) = \eta_x \cdot \sum_y [f_1(x, y) - \tilde{f}_1(x, y)] \cdot Y_1(y) + \alpha_x \cdot \Delta \hat{X}(x)$$

$$\Delta Y_1(y) = -\eta_y \cdot \frac{\partial E}{\partial Y_1(y)} + \alpha_y \cdot \Delta \hat{Y}(y) = \eta_y \cdot \sum_x [f_1(x, y) - \tilde{f}_1(x, y)] \cdot X_1(x) + \alpha_y \cdot \Delta \hat{Y}(y)$$

上式において η_x 、 η_y は修正パラメータであり、第 1 の実施形態で説明したものと同様のものである。

【0040】

数値適用例では、「数10-B」の式を用いて修正量を算出している。図11に示した 1 次元適応化基底組に対する修正量、即ち基底設定処理（ステップ S21）直後の場合は、前回の修正量 $\Delta \hat{X}(x)$ 、 $\Delta \hat{Y}(y)$ は、全ての要素が 0 であるので、慣性項係数の部分は 0 となる。このときの修正量を図13に示す。この図中の式の、括弧内の t の値は、何回目の修正量であることを示している。つまり、 $t = n$ ならば、 n 回目の 1 次元適応化基底の修正量となる。ちなみに、ここで用いた修正パラメータは、 $\eta_x = \eta_y = 0.00078$ 、慣性項係数は、 $\alpha_x = \alpha_y = 0.4$ としている。

【0041】

上記算出した修正量 $\Delta X_1(x)$ 、及び $\Delta Y_1(y)$ を用い、基底修正（ステップ S25）において、基底設定（ステップ S21）で選択された 1 次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ に対して、修正後の各 1 次元適応化基底組を $X'_1(x)$ 、 $Y'_1(y)$ 、修正前

の1次元適応化基底組を $X_i(x)$ 、 $Y_i(y)$ とした時に、各1次元適応化基底組を次のように修正する。

【数11】

$$X'_i(x) = X_i(x) + \Delta X_i(x)$$

$$Y'_i(y) = Y_i(y) + \Delta Y_i(y)$$

【0042】

これにより、図11に示した1次元適応化基底組は、図14のようになる。この1回目の修正では、初期の1次元適応化基底組の要素が、1つを除いて全て0であったため、1部の要素しか修正されていない。また、0以外である数値も、0.1と、比較的小さいため、最初の修正量は極微量なものとなっている。勿論、好適には全ての数値に乱数を入れておくため、1回目の修正から、全ての要素が修正されることになる。しかし、初期の値は、ほぼ0と見なせる値にしておくことが望ましいので、やはり修正量は微量となる。この修正された1次元適応化基底組を用いて、「数6」で示される式に従い誤差Eを算出すると、482911.48という値が得られる。このように、1回目の修正では、誤差Eはわずかに減少する程度である。このような修正を、第1の実施形態と同様に、誤差下限判定(ステップS26)において終了判定、若しくは誤差収束判定(ステップS27)において収束判定がなされるまで繰り返し行うことで、基底設定(ステップS21)で選択した基底番号の1次元適応化基底組が修正される。例えば、図13に示した1回目の修正量により修正した後の、2回目の修正量は、図15に示すようになり、この修正により、誤差Eは482906.05となる。2回目の修正では、まだ修正量が微量であるが、1回目とは異なり、全要素が修正されることになる。この数値適用例の場合、7~10回目の修正量が比較的大きくなり、例えば9回目の修正量は、図16に示すようになる。9回目の修正の、直前の誤差Eは、286696.02であり、9回目の修正の直後は、誤差Eは、16349.27と大きく減少する。この修正回数と誤差Eの変化の関係を図17に示す。図17に示すように、誤差Eは修正を行うごとに、9回目の修正まで加速的に減少し、その後一旦上昇する。そしてその後は振動しながら徐々に収束していく。9回目の修正後の、大きな上昇は、慣性項係数の影響によるところが大きく、慣性項係数を0とすると、このような大きな上昇は起こりにくいが、収束までの修正回数や、局所的な極小値へのトラップを回避するために有効である。

【0043】

誤差下限判定(ステップS26)、及び誤差収束判定(ステップS27)での判定処理は、第1の実施形態で示したものと同様である。誤差下限判定(ステップS26)において、終了判定が出た場合は、ここまでに選択され修正された基底組数と、それらの1次元適応化基底組を符号化データとして記録して処理を終了する。上記の数値適用例では、この段階では誤差がまだ大きく、十分な近似ができていないといえない。例えば、誤差の下限を100としておくと、ここでは終了判定が出ないことになる。終了判定が出なかった場合は、次の誤差収束判定(ステップS27)に進み、ここで誤差Eの収束判定を行う。数値適用例では、30回程度修正を行うと、誤差Eはほぼ収束し、約16199となる。収束後の1次元適応化組を図18に示す。また、この1次元適応化基底組により復元される要素復元画像 $f_0^-(x, y)$ に対応する 8×8 の数値列を図19に示す。図中、1次元適応化基底の修正回数が、 $t \rightarrow \infty$ となっているが、これは、誤差Eが収束する程度の回数修正を行ったという意味である。

【0044】

第1の実施形態では、誤差収束判定(ステップS27)において、誤差Eが収束したと判定された場合は、終了判定を出し処理を終了したが、本第2の実施形態では、次の処理である差分算出(ステップS28)の処理に進む。収束したと判定されない場合は、基底復元(ステップS22)に戻り、選択された基底番号の1次元適応化基底組に対する修正を繰り返し実行する。

【0045】

さて、誤差Eが収束したと判定された場合、差分算出（ステップS28）において、基底番号1の1次元適応化基底組から復元される要素復元画像 $f^{-1}(x, y)$ を

【数12】

$$\tilde{f}_1(x, y) = X_1(x) \cdot Y_1(y)$$

で算出し、差分被符号化画像 $f_{1+1}(x, y)$ を、

【数13】

$$f_{i+1}(x, y) = f(x, y) - \sum_{n=0}^i \tilde{f}_n(x, y)$$

により生成する。上記説明してきた数値適用例では、 $i=0$ であったので、差分被符号化画像 $f_1(x, y)$ は、入力画像 $f(x, y)$ から、基底番号0の1次元適応化基底組から復元される要素復元画像 $f^{-0}(x, y)$ を単純に差し引いたものになる。この差分被符号化画像 $f_1(x, y)$ に対応する 8×8 の数値列は、図20に示したようになる。

【0046】

次に、基底数判定（ステップS28）において、ここまでに修正が完了した基底組数が、予め定めた所定数に達していた場合は、終了判定を出し、ここまでに選択され修正された基底組数と、それらの1次元適応化基底組を符号化データとして記録して処理を終了する。基底組数が所定数（許容数）に達していない場合は、基底設定（ステップS21）に戻り、新たな1次元適応化基底組を選択して、上記説明した基底復元（ステップS22）から誤差収束判定（ステップS27）までの処理を繰り返し実行し、新たに選択された1次元適応化基底組の修正を行う。

【0047】

上記説明してきた数値適用例では、まだ1組しか修正が完了していないので、例えば基底組数の許容数を4と設定しておく、この段階では所定数に達していないので、基底設定（ステップS21）に戻り、次の基底番号1の1次元適応化基底組を選択し、差分算出（ステップS28）において求めた、差分被符号化画像 $f_1(x, y)$ を用い、「数10-B」で示される式に従って、1次元適応化基底組の修正を行っていくことになる。

【0048】

初期の1次元適応化基底組は、上述のように、全て図11に示したものである、ここで選択される基底番号1の1次元適応化基底組も図11に示したものと同一である。この時、まず一回目の修正量は、図21に示したものになり、基底番号0の1回目の修正と同様に、一部の要素しか修正されない。この1回目の修正の、直前の誤差Eは、16199.19であったが、この修正により、誤差Eは16199.14となり、微量ではあるが減少する。そして、2回目の修正量は、図22に示したようになり、修正後の誤差Eは、16199.07となる。このような修正を、誤差が収束するまで繰り返すことで、誤差Eは、約4749程度まで減少して収束する。

【0049】

誤差が収束したら、「数13」の式に基づき、次の差分被符号化画像 $f_2(x, y)$ を求める。ここで求めた、差分被符号化画像 $f_2(x, y)$ は、入力画像 $f(x, y)$ から、基底番号0の1次元適応化基底組から復元される要素復元画像 $f^{-0}(x, y)$ と、基底番号1の1次元適応化基底組から復元される要素復元画像 $f^{-1}(x, y)$ とを差し引いたものになる。

【0050】

このように、差分被符号化画像は、修正の完了した全ての1次元適応化基底組から復元される要素復元画像を、入力画像から差し引いた物となる。この差分被符号化画像 $f_2(x, y)$ に対し、基底番号2の、初期の1次元適応化基底組を選択し、「数10-B」の式の基づき、この1次元適応化基底組の修正を行うというように、徐々に1次元適応化基底組の数を増やしていく。

【0051】

数値適用例では、1組目の1次元適応化基底組の修正が完了した後は、上述のように、誤差Eが約16199となり、2組目の修正が完了すると、誤差Eは、約4749となる。そして、3組目、4組目の修正がそれぞれ完了すると、誤差Eは約3312、約671となる。この数値適用例では、基底組数の許容量を4としていたので、誤差の大きさにかわらず、基底数判定（ステップS28）により、この段階で処理が終了する。もし、誤差下限判定（ステップS26）での誤差の下限を1000と設定していると、4組目の修正の途中で、誤差が設定した下限を下回るので、下回った段階で処理が終了となる。

しかし、基底組数を増やさずに、まだ誤差が減少する余地があるのに、途中で修正を打ち切るのは、処理時間の高速化としては意義があるが、近似性能という意味では得策ではない。そこで、誤差下限判定（ステップS26）での終了判定は、下限を下回った時の1次元適応化基底組の修正において、誤差が収束したら終了判定とする、というようにしても良い。

【0052】

基底数判定処理（ステップS28）において用いる、予め定めた基底組の所定数は、無限個と設定しておいても構わないが、そのようにすると、誤差下限判定（ステップS26）で用いる誤差の許容範囲を仮に小さく設定していると、非常に多数の基底組数を必要とする可能性が高くなる。そこで、基底組数に予め制限を与えておくことにより、データ量の上限を定めて上記符号化を行うことができる。

【0053】

上記方法により符号化した符号化データを復号化画像 $f_{\text{decode}}(x, y)$ に復号するのは、下記に示すように行えばよい。

【数14】

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} [X_n(x) \cdot Y_n(y)]$$

上式のkは選択され修正された基底組数を意味する。

【0054】

このように、本第2の実施形態の符号化方法では、簡単な処理により、符号化された符号化データから、復号化データを得ることができる特徴がある。

【0055】

数値適用例において得られた4組の1次元適応化基底組と、それにより復元される復号化画像に対応する 8×8 の数値列を図23に示す。

【0056】

この数値適用例では、説明を簡単にするため、符号化する画像を 8×8 としたが、この程度のサイズの場合は、データの圧縮という意味ではほとんど意味がない。データの圧縮という意味での画像データの符号化では、もう少しサイズの大きいデータに対して符号化を行う方が好適である。

【0057】

例えば、図24に示したような、 100×100 画素の画像に対して、上記の符号化を行い、それを復元した例を図25に示す。

【0058】

図25では、左から基底組数が、それぞれ1組、2組、5組、10組、20組の場合に復元される復号化画像を示している。ここからわかるように、20組程度になると十分に近似ができていることがわかる。もし被符号化データである 100×100 の画像の、1つの画素値が、4 Byte の浮動小数点であった場合、データ量は、 $4 \text{ Byte} \times 100 \times 100 = 40000 \text{ Byte}$ となる。これに対し、1次元適応化基底の各要素の値を、同じく4 Byte の浮動小数点とした場合、基底組数20組で、 $4 \text{ Byte} \times (100 [X \text{ 方向 } 1 \text{ 次元適応化基底の要素数}] + 100 [Y \text{ 方向 } 1 \text{ 次元適応化基底の要素数}]) \times 20$ (基底組数) $= 16000 \text{ Byte}$ となりデータの圧縮ができていることになる。

【0059】

しかし実際には、図24に示したような通常の画像の場合、1つの画素値は1Byteで表されているので、データ量は1Byte \times 100 \times 100=10000Byteになる。この場合、1次元適応化基底の各要素の値を4Byteの浮動小数点として持つようにすると、逆にデータ量が増えてしまう。しかし、通常の画像のような場合は、実際には1次元適応化基底の各要素の値に4Byteもの精度は必要なく、例えば8bitや6bitといった程度の精度で充分となる。そこで、例えば、各1次元適応化基底の上限と下限のみを4Byteの浮動小数点精度で保持しておき、その上限と下限を用いて、その1次元適応化基底の要素の値を、所望のbit数にまるめるようにすればよい。

【0060】

図26は、左から、それぞれ8bit、6bit、4bitにまるめて符号化した時の、基底組数20組の場合に復元される復号化画像を示している。図26の1番左側の8bitでまるめたもののデータ量は、1Byte[8bit] \times (100[X方向1次元適応化基底の要素数]+100[Y方向1次元適応化基底の要素数])+4Byte \times 4[1基底組に必要な上限・下限保持データ数] \times 20[基底組数]=4320Byteとなり、データ量の圧縮ができています。ちなみに、6bitの場合は3320Byte、4bitの場合は2320Byteとなり、データの圧縮ができることがわかる。

【0061】

上記示した1次元適応化基底の要素の値をまるめる方法に関しては、この方法に限るものではなく、所定の絶対値を用いてまるめるなど、その他の方法を用いても良い。また、1次元適応化基底の要素の値をまるめる場合は、複数の1次元適応化基底組を用いて符号化が完了した後に、値をまるめるようにしても良いが、誤差が蓄積してしまう可能性がある。1組の1次元適応化基底の修正が完了した段階で、値をまるめる処理を行い、そのまるめた後の1次元適応化基底を用いて、次の1次元適応化基底の修正に用いる差分被符号化画像を求めるようにするのが好適である。

【0062】

また、さらにサイズが大きい画像、例えば1280 \times 960の画像等の場合は、上記の符号化方法をそのまま適用しようとすると、1次元適応化基底の要素数が大きいため、誤差の収束が困難となる可能性が高くなる。このような場合には、予め画像を所定のサイズのブロック、例えば50 \times 50のサイズのブロックに分割して、それぞれの分割されたブロックに対して、上記の符号化を行うようにすればよい。分割サイズを50 \times 50とした場合に、この符号化を図24の画像に適用して、復号化した画像を図27に示す。この場合、左上の50 \times 50のブロックでは、基底組数が8組であり、右上、左下、右下のブロックでは、それぞれ基底組数が10組、7組、12組である。この時、各1次元適応化基底の要素は8bitにまるめた。この場合、この符号化後のデータのサイズは、上限、下限保持データも含めて4292Byteとなり、分割せずに100 \times 100のサイズのまま8bitにまるめて符号化したものに比べ、ややデータ量が小さくなっている。このように、入力画像を適当なサイズで分割し、それぞれに対して符号化を行うことで、構造的に単純なブロックでは少ない基底組数で符号化ができるので、データ量を少なくすることができる。ただし、分割サイズをあまり小さくしすぎると、上述の8 \times 8の例のように、1ブロックごとのデータの圧縮効果が少なり、さらに上限・下限保持データが多くなるので、適当な大きさのサイズにするのが良い。

【0063】

分割のサイズに関しては、符号化する画像に対して適応的に決めるようにしても良いが、予め所定のサイズにしておくようにしても構わない。分割サイズを予め決めておくと、分割した際に余りが出てくることがある。例えば画像が1280 \times 960で、分割サイズが50 \times 50であり、分割を左上詰で行っていった場合は、画像右端に水平方向に30画素、画像下端に垂直方向に10画素の余りが出る。このような場合は、例えば画像右端の部分は30 \times 50、画像下端の部分は50 \times 10、画像右下の部分は30 \times 10のサイズとして、符号化を行えばよい。また、余りの部分も全て50 \times 50として、画像外、つまり画像右端の部分なら、左から20画素分、画像下端の部分ならば、下から40画素分、

画像右下の部分ならば左から20画素、下から40画素分を、全て0として符号化を行うようにしても構わない。

【0064】

以上説明したように第2の実施形態では、有限範囲で定義される離散的な2次元濃淡画像 $f(x, y)$ を、複数の離散的な1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ に符号化する。この結果、第1の実施形態と比較して、符号化対象が自然画の場合にも、精度良く対処することが可能となる。

【0065】

<第3の実施形態>

本第3の実施形態では、上記第2の実施形態の変形として、有限範囲で定義される離散的な2次元濃淡画像 $f(x, y)$ を、分割したブロック毎の、複数の離散的な1次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ に変換し、それを符号化する方法を第3の実施形態として説明する。また、装置構成は第1の実施形態と同じであるものとし、その説明は省略する。

【0066】

図3は、本第3の実施形態における処理のフローを示したものである。以下、同図に従って本第3の実施形態における処理手順を説明する。

【0067】

まず、画像入力(ステップS30)において、符号化される離散的な2次元濃淡画像である入力画像 $f(x, y)$ を入力する。第1、第2の実施形態と同様、 x 、 y の定義域は $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$ であり、共に整数である。

【0068】

次に、画像分割処理(ステップS31)において、入力画像をサイズ L_x 、 L_y (L_x 、 L_y は非負整数) の、以下の式で表される複数のウィンドウ $W^{\xi, \zeta}(x, y)$ に分割する。

【数15】

$$W^{\xi, \zeta}(x, y) = f(x + \xi L_x, y + \zeta L_y)$$

ここで、 ξ 、 ζ は0以上の整数であり、各ウィンドウの定義域は $0 \leq x < L_x$ 、 $0 \leq y < L_y$ である。また、 $x + \xi L_x \geq x_{\max}$ 、若しくは、 $y + \zeta L_y \geq y_{\max}$ となるウィンドウでは、 $x + \xi L_x \geq x_{\max}$ 、若しくは、 $y + \zeta L_y \geq y_{\max}$ の範囲を切り捨ててしまい L_x 、 L_y よりサイズの小さいウィンドウとすればよい。

【0069】

次に、ウィンドウ選択処理を行う(ステップS32)。ここでは、画像分割処理(ステップS31)で入力画像を分割して得られたウィンドウから1つのウィンドウを $W^{\xi, \zeta}(x, y)$ を選択する。選択の順序に関しては特に限定は無く、今まで選択されていないウィンドウを1つ選択すればよい。

【0070】

次いで、ウィンドウ選択処理(ステップS32)で選択されたウィンドウ $W^{\xi, \zeta}(x, y)$ に対し、基底組化処理を行う(ステップS33)。ここでは、選択されたウィンドウ $W^{\xi, \zeta}(x, y)$ を複数の離散的な1次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ に変換する。この基底組化での処理は、第2の実施形態で説明した、複数の1次元適応化基底組に符号化する方法と同様の方法を用いればよい。ここで設定する1次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ の定義域は、基本的には $0 \leq x < L_x$ 、 $0 \leq y < L_y$ とする。また、上記説明した $x + \xi L_x \geq x_{\max}$ 、若しくは、 $y + \zeta L_y \geq y_{\max}$ となるウィンドウでは、そのウィンドウのサイズを

【数16】

$$\tilde{L}_x, \tilde{L}_y$$

としたとき、

【数17】

$$0 \leq x < \tilde{L}_x$$

とすればよい。この画像分割処理（ステップS31）における処理は、具体的には、第2の実施の形態において説明した、所定のブロックサイズに分割したものと同様である。

【0071】

基底組化処理（ステップS33）では、第2の実施形態と同様のものであるので、詳細な説明を割愛する。この基底組化処理（ステップS33）により、ウィンドウ選択（ステップS32）で選択された1つのウィンドウ $W^{\xi, \zeta}(x, y)$ の1次元適応化基底組変換が完了したら、次の選択終了判定処理（ステップS34）に進む。

【0072】

選択終了判定処理（ステップS34）では、全てのウィンドウを選択したかどうかを判定して、まだ選択されていないウィンドウがある場合は、ウィンドウ選択（ステップS32）に戻り、未選択のウィンドウを選択して、そのウィンドウに対して上記基底組化（ステップS33）により、複数の1次元適応化基底組への変換を行う。全てのウィンドウを選択した場合は、次のDCT変換処理（ステップS35）に進む。

【0073】

DCT変換処理（ステップS35）では、基底組化処理（ステップS33）で生成された、各ウィンドウ $W^{\xi, \zeta}(x, y)$ の、複数の離散的な1次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ のそれぞれを、DCTベクトル $U^{\xi, \zeta}_1(x)$ 、 $V^{\xi, \zeta}_1(y)$ に離散コサイン変換する。各1次元適応化基底に対する離散コサイン変換は、以下の変換式を用いて行えばよい。

【数18】

$$U^{\xi, \zeta}_1(x) = \sum_{i=0}^{L_x-1} \left[\sqrt{\frac{2}{L_x}} \cdot c_x \cdot \cos \left\{ \frac{x \cdot (2i+1)}{2L_x} \cdot \pi \right\} \cdot X^{\xi, \zeta}_1(i) \right]$$

$$V^{\xi, \zeta}_1(y) = \sum_{i=0}^{L_y-1} \left[\sqrt{\frac{2}{L_y}} \cdot c_y \cdot \cos \left\{ \frac{y \cdot (2i+1)}{2L_y} \cdot \pi \right\} \cdot Y^{\xi, \zeta}_1(i) \right]$$

ここで c_x は、 $x=0$ のとき、 $c_x=1/\sqrt{2}$ である。また、 $x \neq 0$ のとき、 $c_x=1$ である。 c_y も同様で、 $y=0$ のとき、 $c_y=1/\sqrt{2}$ であり、 $y \neq 0$ のとき、 $c_y=1$ である。

【0074】

上記変換により、基底組化処理（ステップS33）で生成され、全ての離散的な1次元適応化基底 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ それぞれを、DCTベクトル $U^{\xi, \zeta}_1(x)$ 、 $V^{\xi, \zeta}_1(y)$ に変換する。

【0075】

次いで、量子化処理（ステップS36）において、DCT変換処理（ステップS35）により得られた全てのDCTベクトル $U^{\xi, \zeta}_1(x)$ 、 $V^{\xi, \zeta}_1(y)$ に対して量子化を行い、量子化基底 $U_q^{\xi, \zeta}_1(x)$ 、 $V_q^{\xi, \zeta}_1(y)$ を生成する。量子化は、DCTベクトルの要素数の等しい1次元の量子化係数ベクトル $Q_x(x)$ 及び $Q_y(y)$ を用いて以下に示すように行う。

【数19】

$$U_q^{\xi, \zeta}_1(x) = \left[\frac{U^{\xi, \zeta}_1(x)}{Q_x(x)} \right]$$

$$V_q^{\xi, \zeta}_1(y) = \left[\frac{V^{\xi, \zeta}_1(y)}{Q_y(y)} \right]$$

【0076】

ここで、

[a]

は、aを超えない最大の整数を意味し、例えば、

[1.5]=1

となる。本実施形態では、上記のように切捨てを行うが、四捨五入をして整数化するような方法でも構わない。

【0077】

この量子化係数ベクトル $Q_x(x)$ 及び $Q_y(y)$ は、J P E G形式の量子化係数と同様、高周波成分、つまり x が大きい成分ほど量子化係数を大きくするのが好ましい。また、量子化係数ベクトル $Q_x(x)$ 及び $Q_y(y)$ の値は、D C Tベクトルのレンジと、各周波数成分で求められる分解能に応じて予め定めておけばよい。

【0078】

最後に、量子化処理（ステップS36）により得られた量子化基底 $U_{q^{\xi, \zeta}}(x)$ 、 $V_{q^{\xi, \zeta}}(y)$ をエントロピー符号化する（ステップS37）。このエントロピー符号化処理では、全ての量子化基底 $U_{q^{\xi, \zeta}}(x)$ 、 $V_{q^{\xi, \zeta}}(y)$ を符号化データとして符号化を行う。ここでのエントロピー符号化の方法としては、例えばハフマン符号化や、算術符号化等の公知の符号化方法を用いればよい。そして、エントロピー符号化された量子化基底と、ウィンドウごとの基底数や、ウィンドウサイズ等の復号化に必要なインデックス情報を、符号化データの一部として記録して処理を終了する。インデックス情報に関しては、各インデックス情報を、予め定めて規格化されたものを用いれば省略することも可能である。

【0079】

本第3の実施形態では、D C T変換処理（ステップS35）において、複数の離散的な1次元適応化基底組 $X^{\xi, \zeta}(x)$ 、 $Y^{\xi, \zeta}(y)$ に対し、そのまま離散コサイン変換を行ったが、特にこれに限るものではなく、各1次元適応化基底組 $X^{\xi, \zeta}(x)$ 、 $Y^{\xi, \zeta}(y)$ を更に分割し、それぞれに対して離散コサイン変換し、量子化して、エントロピー符号化するようにしても構わない。

【0080】

上記方法により符号化した符号化データを復号化画像 $f_{\text{decode}}(x, y)$ に復号するのは、適宜インデックス情報を用いながら、まずエントロピー復号化により量子化基底 $U_{q^{\xi, \zeta}}(x)$ 、 $V_{q^{\xi, \zeta}}(y)$ を生成して逆量子化し、復号化D C Tベクトル $U_{\text{decode}}^{\xi, \zeta}(x)$ 、 $V_{\text{decode}}^{\xi, \zeta}(y)$ を復元して、それに対して離散コサイン変換の逆変換を行うことにより、復号化1次元適応化基底 $X_{\text{decode}}^{\xi, \zeta}(x)$ 、 $Y_{\text{decode}}^{\xi, \zeta}(y)$ を復元する。次いで、復元した復号化1次元適応化基底 $X_{\text{decode}}^{\xi, \zeta}(x)$ 、 $Y_{\text{decode}}^{\xi, \zeta}(y)$ に基づき、復号化ウィンドウ $W_{\text{decode}}^{\xi, \zeta}(x, y)$ を下記に示す式により生成する。

【数20】

$$W_{\text{decode}}^{\xi, \zeta}(x, y) = \sum_{n=0}^{k^{\xi, \zeta}-1} [X_{\text{decode}}^{\xi, \zeta, n}(x) \cdot Y_{\text{decode}}^{\xi, \zeta, n}(y)]$$

【0081】

上式の $k^{\xi, \zeta}$ は、ウィンドウごとの、選択され修正された基底組数である。復号化画像 $f_{\text{decode}}(x, y)$ は、上記生成した復号ウィンドウ $W_{\text{decode}}^{\xi, \zeta}(x, y)$ を合成することで得ることができる。

【0082】

本第3の実施形態では、第2の実施形態と同様、生成された離散的な1次元適応化基底

組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ から要素復元画像を復元して、差分被符号化画像を生成する。量子化のための誤差を吸収するため、順次生成される離散的な 1 次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ に対し、1 次元適応化基底組が生成されるごとに DCT 変換 (ステップ S 35)、及び量子化 (ステップ S 36) を行い、量子化後の量子化基底 $U_q^{\xi, \zeta}_1(x)$ 、 $V_q^{\xi, \zeta}_1(y)$ から、ただちに上記復号化と同様の手法で、逆量子化、離散コサイン変換の逆変換を行い、復号化 1 次元基底 $X_{\text{decode}}^{\xi, \zeta}_1(x)$ 、 $Y_{\text{decode}}^{\xi, \zeta}_1(y)$ を生成し、この復号化 1 次元適応化基底を用いて、要素復元画像を復元して、差分被符号化画像を生成するようにしてもよい。

【0083】

図 24 に示した画像に対してこの符号化を適用し、それを復号化したものを図 28 に示す。この符号化においては、画像を 8×8 のブロックに分割 ($L_x = L_y = 8$) して、符号化を行っている。この例では、上記のように 1 次元適応化基底組が 1 組生成されるごとに、DCT 変換、量子化を行い、それを逆量子化し、離散コサイン変換の逆変換を行って、それを用いて差分被符号化画像を生成するようにしている。これにより、量子化による誤差が次の 1 次元適応化基底組の生成時に吸収されるようになるので、量子化誤差の蓄積を防ぐことができる。 8×8 のブロックに分割した際に出てくる余りに関しては、画像外を 0 として全てが 8×8 のブロックになるようにして符号化を行った。また、量子化係数ベクトル $Q_x(x)$ 及び $Q_y(y)$ は、ともに図 29 に示したものを用了。この場合、復号化に必要な情報 (画像サイズ: 4 Byte \times 2、各ブロックの基底組数: 1 Byte \times ブロック数 [13 \times 13 組]、量子化係数ベクトル: 4 Byte \times 8) を含めて、3790 Byte となり、第 2 の実施形態の 8 bit にまるめたものと同程度以上の画質でありながら、さらにデータ量の削減ができています。この例では、 8×8 のブロックに分割して、符号化を行ったが、例えば 20×20 のブロックに分割する等、その他のサイズで分割するようにしても構わない。

【0084】

上記第 3 の実施形態に示した符号化方法では、第 2 の実施形態でも述べたように、入力画像を複数のウィンドウに分割して基底組化することにより、ウィンドウに対応する基底組数をウィンドウごとに可変にすることができる。そのため、濃度分布が単調なウィンドウ、例えば図 5 に示したような濃度分布であるウィンドウならば、そのウィンドウは少ない基底組数で誤差の少ない符号化をすることができ、ウィンドウが自然画像のような濃度分布であれば多くの基底組を用いて符号化するというように、基底組数の最適化を行うことができる。また、基底組化したものを離散コサイン変換して量子化し、それをエントロピー符号化することで、さらに高効率の符号化が可能になる。

【0085】

以上、第 3 の実施形態では、有限範囲で定義される離散的な 2 次元濃淡画像 $f(x, y)$ を、分割したブロック毎の、複数の離散的な 1 次元適応化基底組 $X_{\text{decode}}^{\xi, \zeta}_1(x)$ 、 $Y_{\text{decode}}^{\xi, \zeta}_1(y)$ に変換し、それを符号化する方法の例として説明した。

【0086】

< 第 4 の実施形態 >

本第 4 の実施形態では、第 3 の実施形態の変形として、有限範囲で定義される離散的な 2 次元濃淡画像 $f(x, y)$ を 2 次元離散コサイン変換して量子化し、量子化周波数画像 $f^{u, v}_1(\xi, \zeta)$ を生成して、それを複数の離散的な 1 次元適応化基底組 $X^{u, v}_1(x)$ 、 $Y^{u, v}_1(y)$ に符号化する方法を説明する。

【0087】

装置構成は第 1 の実施形態と同様であり、以下、本第 4 の実施形態の処理を図 4 のフローチャートに従って説明する。

【0088】

まず、画像入力 (ステップ S 40) において、符号化される離散的な 2 次元濃淡画像である入力画像 $f(x, y)$ を入力する。ここでも x, y は、上記第 2 の実施形態と同様の有限の定義域をもち、整数であるとする。

【0089】

次に、2次元DCT変換を行う(ステップS41)。ここでは、入力画像 $f(x, y)$ を所定サイズ、例えば縦幅8、横幅8の画素サイズの分割画像 $W^{\xi, \zeta}(x, y)$ に分割する。ここでの分割画像 $W^{\xi, \zeta}(x, y)$ は、上記第3の実施形態で示したウィンドウ $W^{\xi, \zeta}(x, y)$ と同様のものである。ステップS41では、それぞれの分割画像 $W^{\xi, \zeta}(x, y)$ に対して2次元離散コサイン変換を行う。

【0090】

次いで、2次元離散コサイン変換されたデータを量子化する(ステップS42)。この量子化処理では、所定の量子化係数 $Q(x, y)$ を用いて量子化し、量子化画像 $W_q^{\xi, \zeta}(x, y)$ を生成する。つまり、JPG形式の符号化における、2次元離散コサイン変換後、量子化というステップと同様の変換を行う。ここでの変換はJPG形式の変換と同様であるので詳細な説明は割愛する。

【0091】

本第4の実施形態では、入力画像 $f(x, y)$ を縦幅8、横幅8のサイズの分割画像 $W^{\xi, \zeta}(x, y)$ に分割して2次元離散コサイン変換を行い、高周波で値が大きくなるような量子化係数 $Q(x, y)$ を用いて量子化するが、特にこれに限定するものではなく、その他のサイズで分割する、若しくは他の量子化係数を用いて量子化しても構わない。

【0092】

次いで、周波数選択処理を行う(ステップS43)。ここでは、周波数パラメータ u, v を選択し、周波数画像 $f^{u, v}(\xi, \zeta)$ を生成する。ここで、周波数パラメータ u, v は0以上の整数であり、本第4の実施形態では2次元DCT変換(ステップS41)で、入力画像 $f(x, y)$ を縦幅8、横幅8のサイズの分割画像 $W^{\xi, \zeta}(x, y)$ に分割したので、7までの整数である。そのためここでは、周波数パラメータ u, v は、ともにそれぞれ0から7の8種類あるので、計 $8 \times 8 = 64$ の組み合わせから、1つの周波数パラメータの組み合わせを選択する。ここで生成する周波数画像 $f^{u, v}(\xi, \zeta)$ は、量子化(ステップS42)において生成された量子化画像 $W_q^{\xi, \zeta}(x, y)$ を用いて以下のように表わされる。

$$f^{u, v}(\xi, \zeta) = W_q^{\xi, \zeta}(x, y)$$

上式に示すとおり、周波数画像 $f^{u, v}(\xi, \zeta)$ は、各量子化画像 $W_q^{\xi, \zeta}(x, y)$ 中の、選択された周波数 u, v の成分を並べたものである。

【0093】

次の基底組化処理(ステップS44)では、選択された周波数パラメータ u, v に対応して生成された周波数画像 $f^{u, v}(\xi, \zeta)$ を、複数の離散的な1次元適応化基底組 $X^{u, v}_1(\xi)$ 、 $Y^{u, v}_1(\zeta)$ に変換する。ここでの処理は第2の実施形態で示した、複数の1次元適応化基底組に符号化する処理と同様であるので説明を割愛する。

【0094】

選択終了判定(ステップS45)では、全ての周波数パラメータ u, v を選択したかどうかを判定する。本第4の実施形態では、周波数パラメータ u, v の組み合わせは、全64通りであるので、それを全て選択して、基底組化(ステップS44)によりそれぞれの周波数パラメータ u, v に対応する周波数画像 $f^{u, v}(\xi, \zeta)$ の基底組化が完了していれば、次のエントロピー符号化(ステップS46)に進む。64通り全ての周波数パラメータ u, v の組み合わせを選択していない場合は、周波数選択(ステップS43)に戻り、まだ選択されていない周波数パラメータ u, v の組み合わせを選択し、基底組化(ステップS44)により、選択された周波数パラメータ u, v に対応する周波数画像 $f^{u, v}(\xi, \zeta)$ の基底組化を行う。

【0095】

エントロピー符号化(ステップS46)では、基底組化(ステップS44)において生成された基底組 $X^{u, v}_1(\xi)$ 、 $Y^{u, v}_1(\zeta)$ をエントロピー符号化し、符号化データとして、復元に必要なインデックス情報とともに記録して処理を終了する。ここでの処理は、上記第3の実施形態で説明したものと同様の処理を行えばよい。

【0096】

この符号化データから、復号化画像を生成する方法は、第3の実施形態と同様に、適宜インデックス情報を用いながら、まずエントロピー復号化により1次元基底組を復元して、その基底組を用いて周波数画像を復元する。この周波数画像から量子化画像を生成して逆量子化し、それに対し2次元離散コサイン変換の逆変換を行うことで復号化画像を復元することができる。

【0097】

第3の実施形態が入力画像を空間的に分割して符号化したのに対し、上記第4の実施形態では、周波数空間で分割して符号化する方法を示した。本第4の実施形態では、周波数空間で分割したものに対して1次元の基底組化を行ったが、周波数空間で分割したものをさらに空間的に分割して基底組化を行っても構わない。また、エントロピー符号化（ステップS46）では、生成された基底組をそのままエントロピー符号化したが、1次元適応化基底組を量子化してからエントロピー符号化しても良いし、第3の実施形態と同様に、もう1度離散コサイン変換（1次元）し、量子化してエントロピー符号化しても構わない。

【0098】

以上、有限範囲で定義される離散的な2次元濃淡画像 $f(x, y)$ を2次元離散コサイン変換して量子化し、周波数画像 $f^{u,v}(\xi, \zeta)$ を生成して、それを複数の離散的な1次元適応化基底組 $X^{u,v}_1(\xi)$ 、 $Y^{u,v}_1(\zeta)$ に符号化する方法の例として説明した。

【0099】

<第5の実施形態>

上記第1乃至第4の実施形態では、全て2次元データに対する符号化について説明したが、本発明の符号化方法はこれに限るものではなく、多次元のデータに関しても適用できる。そこで本第5の実施形態では、第2の実施形態の変形として、有限範囲で定義される離散的な3次元データ $f(x, y, z)$ を、複数の離散的な1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ 、 $Z_1(z)$ に符号化する方法を説明する。

【0100】

符号化される離散的な3次元データとしては、3次元空間内の離散的な密度分布データや、離散的な2次元濃淡画像の離散的な時系列データが考えられる。ここでは、3次元データの各要素（2次元濃淡画像の離散的な時系列データの場合には、所定時刻の2次元濃淡画像の画素）が1バイトで表現されるものとして説明する。

【0101】

まずはこのような符号化される3次元データ $f(x, y, z)$ を入力する。ここでも第2の実施形態と同様に、 x 、 y 、 z は所定の定義域 $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$ 、 $0 \leq z < z_{\max}$ を有し、整数であるとする。この3次元データに対し、下記の誤差 E 、修正量 $\Delta X_1(x)$ 、 $\Delta Y_1(y)$ 、 $\Delta Z_1(z)$ に基づいて、第2の実施形態と同様に1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ 、 $Z_1(z)$ を修正していくことで符号化を行う。

【数 2 1】

$$E = \sum_x \sum_y \sum_z \left[\frac{1}{2} \{f_l(x, y, z) - \tilde{f}_l(x, y, z)\}^2 \right]$$

$$\Delta X_l(x) = -\eta_x \cdot \frac{\partial E}{\partial X_l(x)} = \eta_x \cdot \sum_y \sum_z [\{f_l(x, y, z) - \tilde{f}_l(x, y, z)\} \cdot Y_l(y) \cdot Z_l(z)]$$

$$\Delta Y_l(y) = -\eta_y \cdot \frac{\partial E}{\partial Y_l(y)} = \eta_y \cdot \sum_x \sum_z [\{f_l(x, y, z) - \tilde{f}_l(x, y, z)\} \cdot X_l(x) \cdot Z_l(z)]$$

$$\Delta Z_l(z) = -\eta_z \cdot \frac{\partial E}{\partial Z_l(z)} = \eta_z \cdot \sum_x \sum_y [\{f_l(x, y, z) - \tilde{f}_l(x, y, z)\} \cdot X_l(x) \cdot Y_l(y)]$$

上式中の $f_l(x, y, z)$ や、 $\tilde{f}_l(x, y, z)$ は、第 2 の実施形態での差分被符号化画像 $f_l(x, y)$ 、要素復元画像 $\tilde{f}_l(x, y)$ と同意であり、次元を 3 次元としただけのものである。その他のパラメータ等に関しても、第 2 の実施形態と同様のものであるので説明は割愛する。

【0 1 0 2】

上記誤差 E 、修正量 $\Delta X_l(x)$ 、 $\Delta Y_l(y)$ 、 $\Delta Z_l(z)$ に基づき、第 2 の実施形態で説明したのと同様の方法で、1 次元適応化基底組に対して修正をしていくことで、3 次元データを 1 次元適応化基底組 $X_l(x)$ 、 $Y_l(y)$ 、 $Z_l(z)$ に符号化する。各基底の修正に関しては、選択された 1 次元適応化基底組 $X_l(x)$ 、 $Y_l(y)$ 、 $Z_l(z)$ に対して、修正後の各 1 次元適応化基底組を $X'_l(x)$ 、 $Y'_l(y)$ 、 $Z'_l(z)$ としたとき、次のように修正する。

【数 2 2】

$$X'_l(x) = X_l(x) + \Delta X_l(x)$$

$$Y'_l(y) = Y_l(y) + \Delta Y_l(y)$$

$$Z'_l(z) = Z_l(z) + \Delta Z_l(z)$$

【0 1 0 3】

最後に、誤差 E が所定値以下になるか、若しくは基底数が所定数に達した場合に、今までに選択して修正した 1 次元適応化基底組 $X_l(x)$ 、 $Y_l(y)$ 、 $Z_l(z)$ を記録して処理を終了する。ここでの処理に関しても、第 2 の実施形態と同様であるので、詳細な説明は割愛する。この場合、1 次元適応化基底組 $X_l(x)$ 、 $Y_l(y)$ 、 $Z_l(z)$ から復元される復号化データ $f_{\text{decode}}(x, y, z)$ は以下ようになる。

【数 2 3】

$$f_{\text{decode}}(x, y, z) = \sum_{n=0}^{k-1} [X_n(x) \cdot Y_n(y) \cdot Z_n(z)]$$

上式中、 k は基底組数である。このように符号化されるデータの次元数が増えた場合でも、上記のように誤差と修正量を求めることで符号化が可能になる。本実施形態では、入力された 3 次元データをそのまま 1 次元適応化基底に符号化しているが、上記第 3 の実施形態等で示したように、所定範囲で分割したり、離散コサイン変換等により周波数空間で分割したりしても構わない。

【0 1 0 4】

以上、有限範囲で定義される離散的な 3 次元データ $f(x, y, z)$ を、複数の離散的な 1 次元適応化基底組 $X_l(x)$ 、 $Y_l(y)$ 、 $Z_l(z)$ に符号化する方法の例として説明した。

【0105】

<第6の実施形態>

上記第1乃至第5の実施形態では、誤差評価関数として、各要素の差の2乗誤差を用いたが、本発明の符号化方法はこれに限るものではなく、その他の誤差評価関数を用いても適用可能である。そこで本発明の第6の実施形態では、第2の実施形態の変形として、aを参照データ、つまり符号化されるデータとし、bを評価データ、つまり符号化されたデータから復元されるデータとした時に、誤差評価関数として $\sum \{1/2 \times (a^2 - ab)^2\}$ を用い、有限範囲で定義される離散的な2次元濃淡画像 $f(x, y)$ を、複数の離散的な1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ に符号化する方法を説明する。

【0106】

まず、符号化される離散的な2次元濃淡画像 $f(x, y)$ を入力する。ここでも第2の実施形態と同様に、 x, y は所定の定義域 $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$ を有し、整数であるとする。この2次元濃淡画像 $f(x, y)$ に対し、下記の誤差 E 、修正量 $\Delta X_1(x)$ 、 $\Delta Y_1(y)$ に基づいて、第2の実施形態と同様に1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ を修正していくことで符号化を行う。

【数24】

$$E = \sum_x \sum_y \left[\frac{1}{2} \{f_i(x, y)^2 - f_i(x, y) \cdot \tilde{f}_i(x, y)\}^2 \right]$$

$$\Delta X_i(x) = -\eta_x \cdot \frac{\partial E}{\partial X_i(x)} = \eta_x \cdot \sum_y [f_i(x, y) \cdot \{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot Y_i(y)]$$

$$\Delta Y_i(y) = -\eta_y \cdot \frac{\partial E}{\partial Y_i(y)} = \eta_y \cdot \sum_x [f_i(x, y) \cdot \{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot X_i(x)]$$

【0107】

上式中の $f_i(x, y)$ や、 $f^{-1}_i(x, y)$ は、第2の実施形態での差分被符号化画像 $f_1(x, y)$ 、要素復元画像 $f^{-1}_1(x, y)$ と同様のものである。また、その他のパラメータ等に関しても、第2の実施形態と同様のものであるので説明は割愛する。

【0108】

上記誤差 E 、修正量 $\Delta X_1(x)$ 、 $\Delta Y_1(y)$ に基づき、第2の実施形態で説明したのと同様の方法で、1次元適応化基底組を修正をしていくことで、1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ に符号化する。各1次元適応化基底組の修正に関しては、選択された1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ に対して、修正後の各1次元適応化基底組を $X'_1(x)$ 、 $Y'_1(y)$ 、修正前の基底を $X_1(x)$ 、 $Y_1(y)$ とした時に、次のように修正する。

【数25】

$$X'_1(x) = X_1(x) + \Delta X_1(x)$$

$$Y'_1(y) = Y_1(y) + \Delta Y_1(y)$$

【0109】

最後に、誤差 E が所定値以下になるか、若しくは基底組数が所定数に達した場合に、今までに選択して修正した1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ を記録して処理を終了する。ここでの処理に関しても、第2の実施形態と同様であるので、詳細な説明は割愛する。この場合の復号化画像 $f_{\text{decode}}(x, y)$ は、第2の実施形態で示したものと同様である。

【0110】

このように、2乗誤差以外の誤差評価関数を用いても、上記のように誤差と修正量を求めることで符号化が可能になる。本第6の実施形態では、誤差評価関数として、 $\sum \{1/$

$2 \times (a^2 - a b)^2$ を用いたが、本発明の符号化方法はこれに限るものではなく、被符号化データに近づくことにより誤差が減少するようなものであれば、その他の誤差評価関数であっても構わない。例えば 2 次元画像であれば、差分被符号化画像と、要素復元画像を微分したものの差分を誤差評価関数に含むようにする等しても良い。そのような誤差評価関数のうち、上記示したような、修正すべきパラメータ、ここでは $X_1(x)$ 、 $Y_1(y)$ での偏微分が解析的に求められることが好ましいが、解析的に偏微分を求められない場合は、修正すべきパラメータに微小変化を与えた時の誤差の増減から修正量を決めてやるようにすればよい。

【0 1 1 1】

以上、誤差評価関数に $\Sigma 1/2 \times \{a^2 - a b\}^2$ を用い、有限範囲で定義される離散的な 2 次元濃淡画像 $f(x, y)$ を、複数の離散的な 1 次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ に符号化する方法の例として説明した。なお、本第 6 の実施形態では、特に第 2 の実施形態に適用する例を説明したが、他の実施形態にも同様に適用できるのは明らかである。

【0 1 1 2】

< 第 7 の実施形態 >

上記第 1 乃至第 6 の実施形態では、多次元のデータを複数の 1 次元適応化基底組に符号化する方法を示したが、本発明の符号化方法はこれに限るものではなく、その他のデータに符号化する方法においても適用可能である。そこで本発明の第 7 の実施形態では、第 2 の実施形態の変形として、有限範囲で定義される離散的な 2 次元濃淡画像 $f(x, y)$ を、複数の離散的な 2 次元適応化基底 $S_1(x_s, y_s)$ に符号化する方法を説明することとする。

【0 1 1 3】

まず、符号化される離散的な 2 次元濃淡画像 $f(x, y)$ を入力する。ここでも第 2 の実施形態と同様に、 x, y は所定の定義域 $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$ を有し、整数であるとする。次に基底として、離散的な 2 次元適応化基底 $S_1(x_s, y_s)$ を 1 つ選択する。ここで x_s, y_s は、 x_{\max} もしくは y_{\max} が偶数である場合は、それぞれの定義域が $0 \leq x_s < (x_{\max}/2) + 1$ 、 $0 \leq y_s < (y_{\max}/2) + 1$ となる整数である。 x_{\max} 、 y_{\max} が奇数である場合は、それぞれの定義域は $0 \leq x_s < (x_{\max}/2) + 0.5$ 、 $0 \leq y_s < (y_{\max}/2) + 0.5$ となる整数である。また、添字 1 は基底番号であり、特に限定はしないが、本第 7 の実施形態でも 0 から順に、予め定めた基底数まで順に選択していくことにする。

【0 1 1 4】

次に、選択した基底番号 1 の 2 次元適応化基底 $S_1(x_s, y_s)$ から、要素復元画像 $f^{-1}_1(x, y)$ を次のように算出する。

【数 2 6】

i) x, y : 偶数

$$\tilde{f}_i(x, y) = S_i\left(\frac{x}{2}, \frac{y}{2}\right)$$

ii) x, y : 奇数

$$\tilde{f}_i(x, y) = \frac{1}{4} \left\{ S_i\left(\frac{x-1}{2}, \frac{y-1}{2}\right) + S_i\left(\frac{x+1}{2}, \frac{y-1}{2}\right) + S_i\left(\frac{x-1}{2}, \frac{y+1}{2}\right) + S_i\left(\frac{x+1}{2}, \frac{y+1}{2}\right) \right\}$$

iii) x : 偶数、 y : 奇数

$$\tilde{f}_i(x, y) = \frac{1}{2} \left\{ S_i\left(\frac{x}{2}, \frac{y-1}{2}\right) + S_i\left(\frac{x}{2}, \frac{y+1}{2}\right) \right\}$$

iv) x : 奇数、 y : 偶数

$$\tilde{f}_i(x, y) = \frac{1}{2} \left\{ S_i\left(\frac{x-1}{2}, \frac{y}{2}\right) + S_i\left(\frac{x+1}{2}, \frac{y}{2}\right) \right\}$$

【0115】

上記求めた要素復元画像 $f^{-1}(x, y)$ を用いて、誤差 E 、及び修正量 $\Delta S_i(x_s, y_s)$ を以下のように算出する。

【数 2 7】

$$E = \sum_x \sum_y \left[\frac{1}{2} \{ f_i(x, y) - \tilde{f}_i(x, y) \}^2 \right]$$

$$\Delta S_i(x_s, y_s) = -\eta \cdot \frac{\partial E}{\partial S_i(x_s, y_s)} = \eta \cdot \sum_{n=-1}^1 \sum_{m=-1}^1 \left[\{ f_i(2x_s + n, 2y_s + m) - \tilde{f}_i(2x_s + n, 2y_s + m) \} \cdot C(m, n) \right]$$

【0116】

上式の $f_i(x, y)$ は差分被符号化画像であり、第2の実施形態と同様のものである。また修正量 $\Delta S_i(x_s, y_s)$ を算出する式において $C(m, n)$ は次の値を有する。

・ $n=0$ 、且つ、 $m=0$ の場合

$$C(n, m) = 1$$

・ $n=\pm 1$ 、且つ、 $m=0$ の場合、又は、 $n=0$ 、且つ、 $m=\pm 1$ の場合

$$C(n, m) = 0.5$$

・ $n=\pm 1$ 、且つ、 $m=\pm 1$ の場合

$$C(n, m) = 0.25$$

ただし、上記算出式において、 $2x_s + n < 0$ 、 $2x_s + n \geq x_{\max}$ 、 $2y_s + m < 0$ 、又は、 $2y_s + m \geq y_{\max}$ のいずれかに場合には、 $C(n, m) = 0$ とする。

【0117】

また η は修正パラメータであり、第2の実施形態で説明した η_x 、 η_y と同様のものである。上記誤差 E 、修正量 $\Delta S_i(x_s, y_s)$ に基づき、第2の実施形態で説明したのと同様の方法で、2次元適応化基底に対して修正をしていくことで、入力画像 $f(x, y)$ を2次元適応化基底 $S_i(x_s, y_s)$ に符号化する。

【0118】

基底の修正に関しては、選択された2次元適応化基底 $S_i(x_s, y_s)$ を次のように修正する。式中、 $S'_i(x_s, y_s)$ が修正後、 $S_i(x_s, y_s)$ が修正前である。

$$S'_i(x_s, y_s) = S_i(x_s, y_s) + \Delta S_i(x_s, y_s)$$

【0119】

最後に、誤差 E が所定値以下になるか、若しくは基底数が所定数に達した場合に、今までに選択して修正した 2 次元適応化基底 $S_1(x_s, y_s)$ を記録して処理を終了する。この処理に関しても、第 2 の実施形態と同様であるので、詳細な説明は割愛する。この方法により符号化した符号化データを復号化画像 $f_{\text{decode}}(x, y)$ に復号するのは、下記に示すように行えばよい。

【数 28】

i) x, y : 偶数

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} S_n\left(\frac{x}{2}, \frac{y}{2}\right)$$

ii) x, y : 奇数

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} \left[\frac{1}{4} \left\{ S_n\left(\frac{x-1}{2}, \frac{y-1}{2}\right) + S_n\left(\frac{x+1}{2}, \frac{y-1}{2}\right) + S_n\left(\frac{x-1}{2}, \frac{y+1}{2}\right) + S_n\left(\frac{x+1}{2}, \frac{y+1}{2}\right) \right\} \right]$$

iii) x : 偶数、 y : 奇数

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} \left[\frac{1}{2} \left\{ S_n\left(\frac{x}{2}, \frac{y-1}{2}\right) + S_n\left(\frac{x}{2}, \frac{y+1}{2}\right) \right\} \right]$$

iv) x : 奇数、 y : 偶数

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} \left[\frac{1}{2} \left\{ S_n\left(\frac{x-1}{2}, \frac{y}{2}\right) + S_n\left(\frac{x+1}{2}, \frac{y}{2}\right) \right\} \right]$$

【0120】

上式の k は選択され修正された基底数である。本第 7 の実施形態では、生成する 2 次元適応化基底 $S_1(x_s, y_s)$ を、要素数が入力画像 $f(x, y)$ の約 $1/4$ の画像としたが、これに限るものではなく、これとは異なる要素数の 2 次元適応化基底を用いて符号化しても構わない。

【0121】

以上、有限範囲で定義される離散的な 2 次元濃淡画像 $f(x, y)$ を、複数の離散的な 2 次元適応化基底 $S_1(x_s, y_s)$ に符号化する第 7 の実施形態を説明した。

【0122】

<第 8 の実施形態>

上記第 1 乃至第 7 の実施形態では、適応化基底として 1 次元適応化基底や 2 次元適応化基底を用いて符号化する方法を示したが、本発明の符号化方法はこれに限るものではなく、いくつかのパラメータで構成されるような適応化基底を用いるような場合でも適用可能である。そこで本発明の第 8 の実施形態では、第 2 の実施形態の変形として、有限範囲で定義される離散的な 2 次元濃淡画像 $f(x, y)$ を、複数のパラメータで構成される適応化基底に符号化する例を説明する。以下では、具体的な例として、4 つのパラメータ A_1 、 p_1 、 q_1 、 ρ_1 から構成される適応化基底 $G_1(A_1, p_1, q_1, \rho_1)$ に符号化する方法を説明する。

【0123】

まず、符号化される離散的な 2 次元濃淡画像 $f(x, y)$ を入力する。ここでも第 2 の実施形態と同様に、 x, y は所定の定義域 $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$ を有し、整数であるとする。次に初期の適応化基底として、適応化基底 G_1 を 1 つ選択する。添字 1 は基底番号であり、特に限定はしないが、本第 8 の実施形態でも 0 から順に、予め定めた基底数まで順に選択していくことにする。

【0124】

次に、選択した基底番号1の適応化基底 G_1 から、要素復元画像 $f^{-1}(x, y)$ を次のように算出する。

【数29】

$$\tilde{f}_i(x, y) = A_i \cdot \exp\left\{\frac{(x-p_i)^2 + (y-q_i)^2}{2\sigma_i^2}\right\}$$

【0125】

次に、上記求めた要素復元画像 $f^{-1}(x, y)$ を用いて、誤差 E 、及び修正量 ΔA_1 、 Δp_1 、 Δq_1 、 $\Delta \rho_1$ を以下のように算出する。

【数30】

$$E = \sum_x \sum_y \left[\frac{1}{2} \{f_i(x, y) - \tilde{f}_i(x, y)\}^2 \right]$$

$$\Delta A_i = -\eta_A \cdot \frac{\partial E}{\partial A_i} = -\eta_A \cdot \sum_x \sum_y \left[\{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot \frac{\tilde{f}_i(x, y)}{A_i} \right]$$

$$\Delta p_i = -\eta_p \cdot \frac{\partial E}{\partial p_i} = -\eta_p \cdot \sum_x \sum_y \left[\{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot \tilde{f}_i(x, y) \cdot \frac{x-p_i}{\sigma_i^2} \right]$$

$$\Delta q_i = -\eta_q \cdot \frac{\partial E}{\partial q_i} = -\eta_q \cdot \sum_x \sum_y \left[\{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot \tilde{f}_i(x, y) \cdot \frac{y-q_i}{\sigma_i^2} \right]$$

$$\Delta \sigma_i = -\eta_\sigma \cdot \frac{\partial E}{\partial \sigma_i} = -\eta_\sigma \cdot \sum_x \sum_y \left[\{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot \tilde{f}_i(x, y) \cdot \frac{(x-p_i)^2 + (y-q_i)^2}{\sigma_i^3} \right]$$

【0126】

上式の $f_1(x, y)$ は差分被符号化画像であり、第2の実施形態と同様のものである。また η_A 、 η_p 、 η_q 、 η_σ は修正パラメータであり、第2の実施形態で説明した η_x 、 η_y と同様のものである。上記誤差 E 、修正量 ΔA_1 、 Δp_1 、 Δq_1 、 $\Delta \rho_1$ に基づき、第2の実施形態で説明したと同様の方法で、適応化基底を構成する各パラメータに対して修正をしていくことで、適応化基底 G_1 に符号化する。

【0127】

適応化基底の修正に関しては、選択された基底 G_1 を構成する各パラメータ A_1 、 p_1 、 q_1 、 ρ_1 を以下のようにして修正する。 A'_1 、 p'_1 、 q'_1 、 ρ'_1 が修正後、 A_1 、 p_1 、 q_1 、 ρ_1 が修正前である。

$$A'_1 = A_1 + \Delta A_1$$

$$p'_1 = p_1 + \Delta p_1$$

$$q'_1 = q_1 + \Delta q_1$$

$$\rho'_1 = \rho_1 + \Delta \rho_1$$

【0128】

最後に、誤差 E が所定値以下になるか、若しくは基底数が所定数に達した場合に、今までに選択して修正した適応化基底 G_1 を構成する各パラメータ A_1 、 p_1 、 q_1 、 ρ_1 を記録出力して処理を終了する。ここでの処理に関しても、第2の実施形態と同様であるので、詳細な説明は割愛する。この方法により符号化した符号化データを復号化画像 $f_{decode}(x, y)$ に復号するのは、下記に示すように行えばよい。

【数 3 1】

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} \left[A_n \cdot \exp \left\{ \frac{(x-p_n)^2 + (y-q_n)^2}{2\sigma_n^2} \right\} \right]$$

上式の k は選択され修正された基底数である。本第 8 の実施形態では、4 つのパラメータにより構成される適応化基底 G_1 を用いたが、これに限るものではなく、これとは異なるパラメータ数の適応化基底を用いて符号化しても構わない。

【0 1 2 9】

以上、有限範囲で定義される離散的な 2 次元濃淡画像 $f(x, y)$ を、複数のパラメータから構成される適応化基底 G_1 に符号化する方法の例として説明した。

【0 1 3 0】

< 第 9 の実施形態 >

本第 9 の実施形態では、所定の 2 次元荷重分布データによる、2 次元分布データに対する離散的コンボリューション演算において、2 次元荷重分布データを第 2 の実施形態のデータ符号化処理により 1 次元適応化基底組に変換し、それを用いて 2 次元分布データに対する離散的コンボリューション演算を行う方法を説明する。ここで示す離散的コンボリューション演算は、例えば、Convolutional Neural Networks (LeCun, Y. and Bengio, Y., 1995, "Convolutional Networks for Images Speech, and Time Series" in Handbook of Brain Theory and Neural Networks (M. Arbib, Ed.), MIT Press, pp. 255-258) における 1 つの特徴検出細胞面の演算等に用いることができる。この場合、2 次元荷重分布データは、シナプス荷重分布と見なせる。

【0 1 3 1】

まず、離散的なコンボリューション演算を行う離散的な 2 次元荷重分布データ $w(x, y)$ を入力する。この 2 次元荷重分布データは離散的であるので、 x, y は整数とする。この 2 次元荷重分布データによる、2 次元分布データ $f(x, y)$ に対する離散的コンボリューション演算結果 $Z(x, y)$ は以下の式で表される。

【数 3 2】

$$Z(x, y) = \sum_u \sum_v \{w(u, v) \cdot f(u+x, v+y)\}$$

【0 1 3 2】

次に、入力された 2 次元荷重分布データ $w(x, y)$ から、第 2 の実施形態で示した方法と同様の方法により、1 次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ を生成する。1 次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ の添え字 1 は基底番号である。ここでの詳細な説明は、第 2 の実施形態で示したものと同様なので説明を割愛する。ここで生成した 1 次元適応化基底組は、コンボリューション演算を行うたびに生成する必要はなく、予め第 2 の実施形態で示した方法により生成しておき、生成した 1 次元適応化基底組を記録して保存しておけばよい。

【0 1 3 3】

生成された 1 次元適応化基底組により、2 次元分布データ $f(x, y)$ に対する離散的コンボリューション演算結果 $Z(x, y)$ は、以下のようになる。

【数 3 3】

$$Z(x, y) = \sum_l \left[\sum_u \sum_v \{X_l(u) \cdot Y_l(v) \cdot f(u+x, v+y)\} \right]$$

【0 1 3 4】

このように、2 次元荷重分布データ $w(x, y)$ を 1 次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ に分解し、それを用いて離散的コンボリューション演算を行うことで、従来は図 6 に示すように、例えば 6×6 の要素で構成される 2 次元荷重分布データによる離散的コン

ポリューション演算であれば、各座標の計 3 6 の要素に独立に 2 次元荷重分布データの値を与える、つまり各座標に対して独立に 2 次元荷重分布データの値を信号として送るために 3 6 本の信号線が必要であったが、本実施形態では図 7 に示すように、計 1 2 本の信号があれば良く、コンポリューション演算器において問題となっていた配線問題を解決することができる。なお、この離散的コンポリューション演算方法を実行するための、積和演算素子をアレイ状に配置したコンポリューション演算器では、図 7 に示すように 1 次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ をアレイの全ての列・行に入力して演算を実行しても良いし、または図 9 に示すように 1 次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ をそれぞれ一部の列・行にのみ入力して、 $X_1(x)$ と $Y_1(y)$ の両方が入力されている、アレイ中の一部の領域 (6 3) においてのみ演算を実行しても構わない。

【0 1 3 5】

以上、所定の 2 次元荷重分布データによる、2 次元分布データに対する離散的コンポリューション演算において、2 次元荷重分布データを第 2 の実施形態のデータ符号化方法により、1 次元適応化基底組に変換し、それを用いて 2 次元分布データに対する離散的コンポリューション演算を行う方法の例として説明した。

【0 1 3 6】

以上、本発明に係る実施形態を説明したが、本発明は、図 8 に示すようにコンピュータプログラムによっても実現できることは明らかであるので、本発明はこのようなコンピュータプログラムをもその範疇とする。また、通常コンピュータプログラムは、CDROM 等のコンピュータ可読記憶媒体に記憶されており、それをコンピュータが有する記憶媒体ドライブ装置にセットし、システムにコピーもしくはインストールすることで実行可能になるわけであるから、本発明はかかるコンピュータ可読記憶媒体をもその範疇とするのは明らかである。

【図面の簡単な説明】

【0 1 3 7】

【図 1】 第 1 の実施形態における符号化処理を示すフローチャートである。

【図 2】 第 2 の実施形態における符号化処理を示すフローチャートである。

【図 3】 第 3 の実施形態における符号化処理を示すフローチャートである。

【図 4】 第 4 の実施形態における符号化処理を示すフローチャートである。

【図 5】 単純な矩形形状の 2 次元画像の例を示す図である。

【図 6】 従来の 2 次元荷重分布データの送信構造を示す図である。

【図 7】 第 9 の実施形態の積和演算方法を実行するコンポリューション演算器における、2 次元荷重分布データの送信構造を示す図である。

【図 8】 実施形態における装置構成を示すブロック構成図である。

【図 9】 第 9 の実施形態の積和演算方法を実行するコンポリューション演算器における、2 次元荷重分布データの送信構造を示す図である。

【図 10】 数値適用例における入力画像に対応する 8×8 の輝度値データを示す図である。

【図 11】 数値適用例における初期の 1 次元適応化基底組を示す図である。

【図 12】 数値適用例における初期の 1 次元適応化基底組から復元される要素復元画像に対応する 8×8 の数値列を示す図である。

【図 13】 数値適用例における基底番号 0 の 1 次元適応化基底組の 1 回目の修正量を示す図である。

【図 14】 数値適用例における基底番号 0 の 1 回目の修正後の 1 次元適応化基底組を示す図である。

【図 15】 数値適用例における基底番号 0 の 1 次元適応化基底組の 2 回目の修正量を示す図である。

【図 16】 数値適用例における基底番号 0 の 1 次元適応化基底組の 9 回目の修正量を示す図である。

【図 17】 数値適用例における基底番号 0 の 1 次元適応化基底組の修正時の修正回数

に対する誤差 E の変化を示す図である。

【図 18】数値適用例における基底番号 0 の誤差 E が収束した後の 1 次元適応化基底組を示す図である。

【図 19】数値適用例における基底番号 0 の誤差 E が収束した後の 1 次元適応化基底組から復元される要素復元画像に対応する 8×8 の数値列を示す図である。

【図 20】数値適用例における基底番号 0 の 1 次元適応化基底組の修正完了後の差分被符号化画像に対応する 8×8 の数値列を示す図である。

【図 21】数値適用例における基底番号 1 の 1 次元適応化基底組の 1 回目の修正量を示す図である。

【図 22】数値適用例における基底番号 1 の 1 次元適応化基底組の 2 回目の修正量を示す図である。

【図 23】数値適用例における修正完了後の 4 組の 1 次元適応化基底組と、それらから復元される復号化画像に対応する 8×8 の数値列を示す図である。

【図 24】第 2 の実施形態における入力画像の例を示す図である。

【図 25】第 2 の実施形態における復号化画像の例を示す図である。

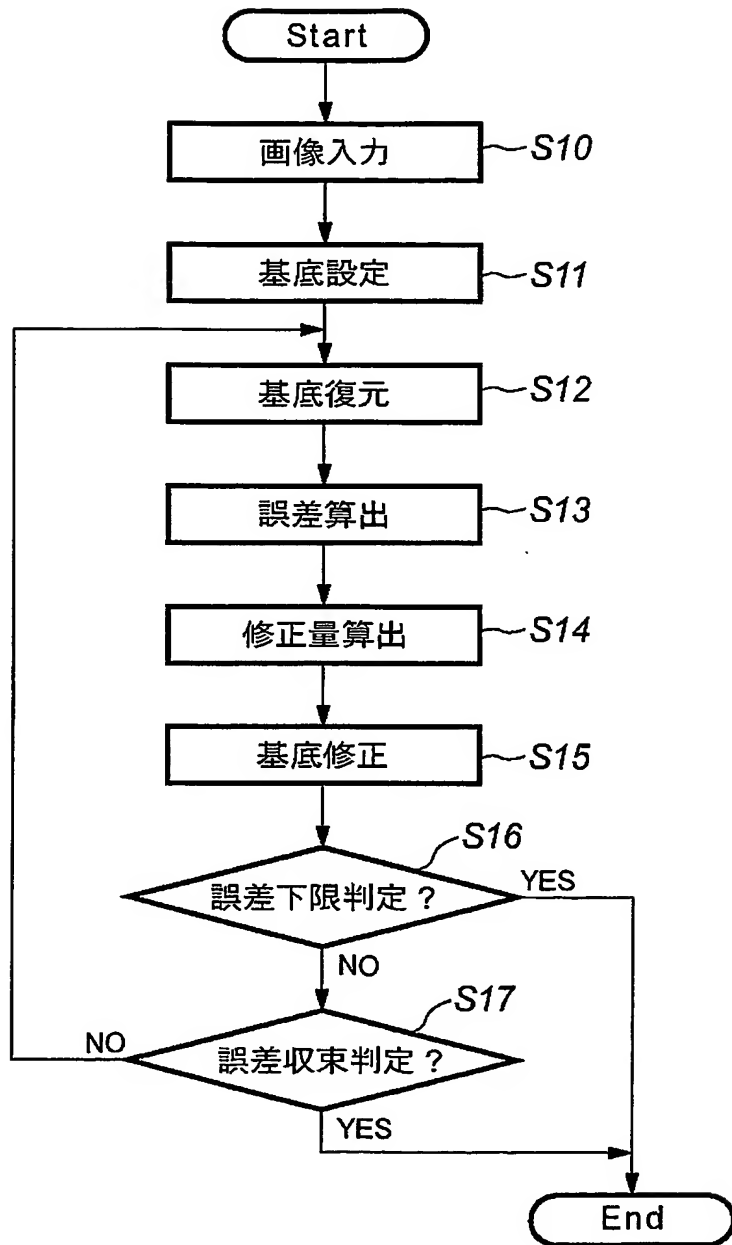
【図 26】第 2 の実施形態における精度をまとめた場合の復号化画像の例を示す図である。

【図 27】第 2 の実施形態における入力画像を分割して符号化し、それを復号化した復号化画像の例を示す図である。

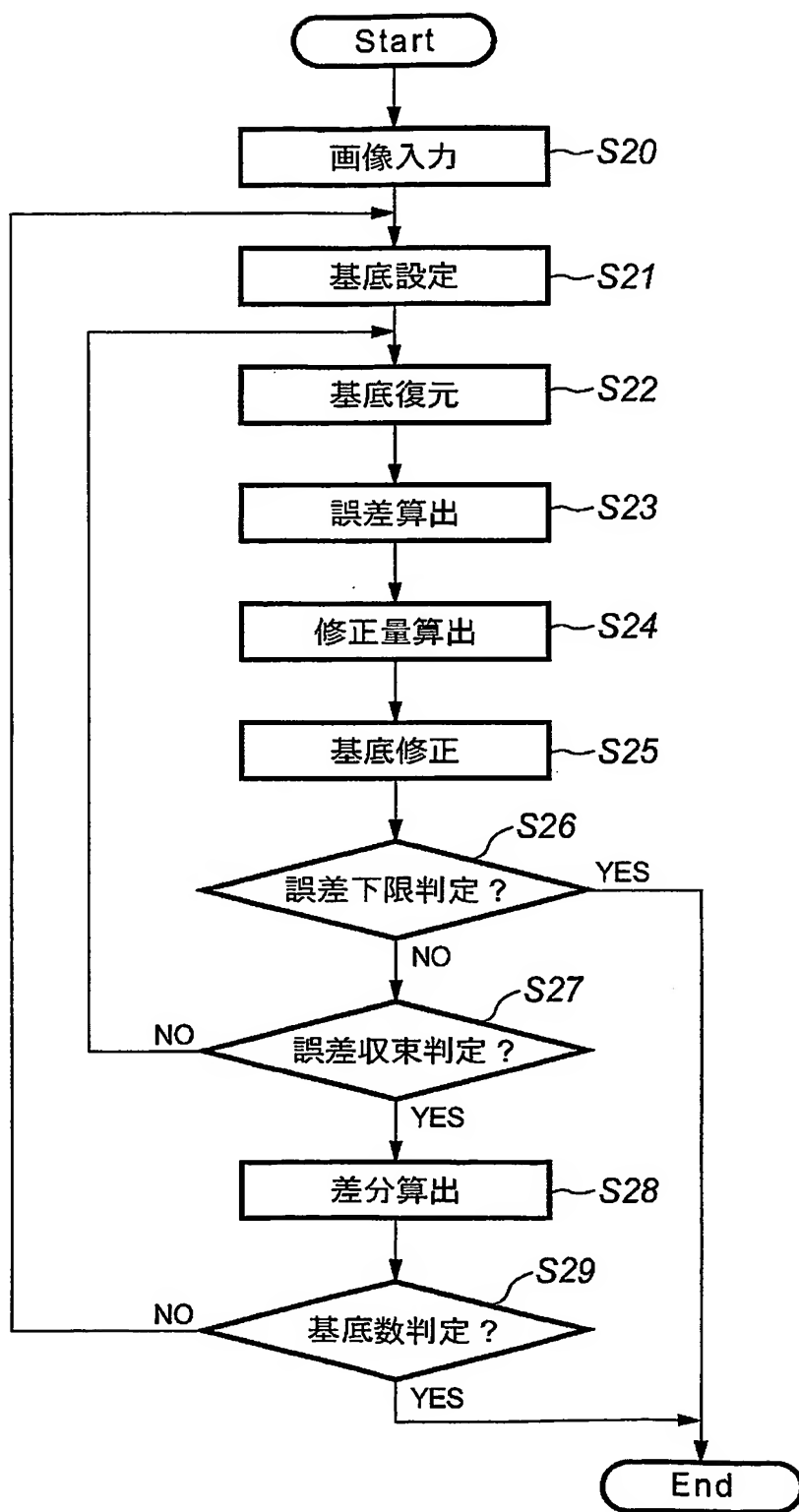
【図 28】第 3 の実施形態における復号化画像の例を示す図である。

【図 29】第 3 の実施形態における量子化係数ベクトルの例を示す図である。

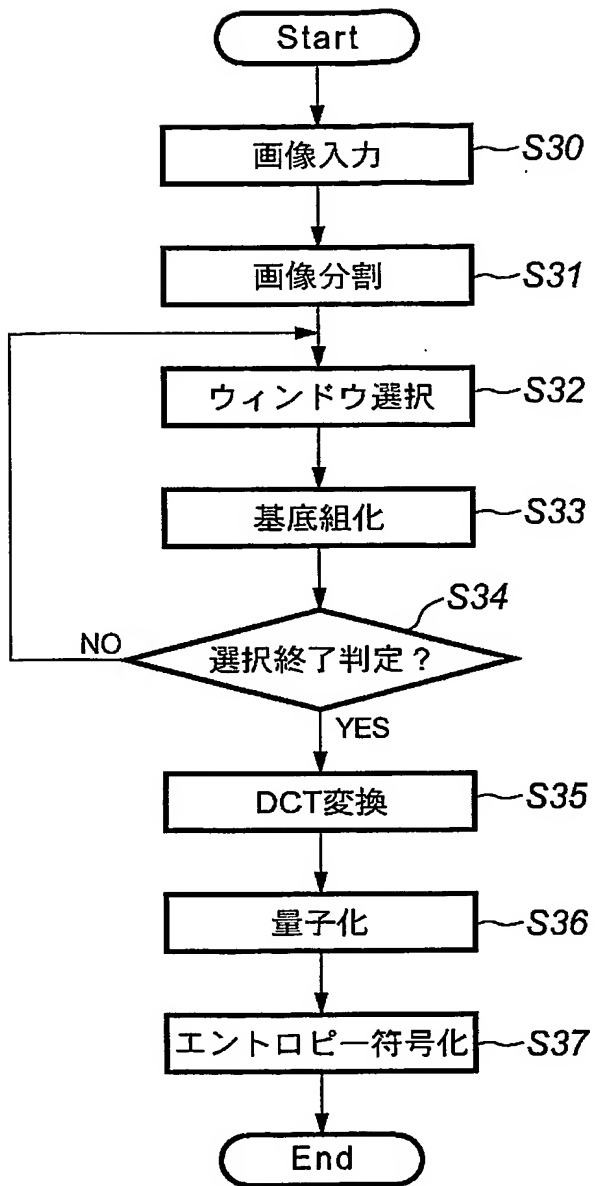
【書類名】図面
【図 1】



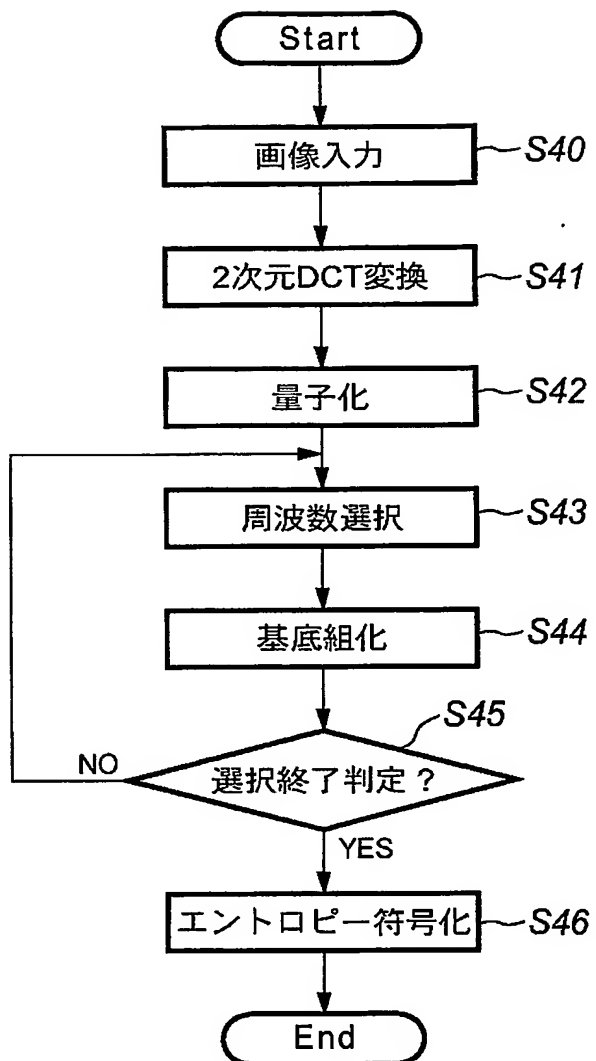
【図 2】



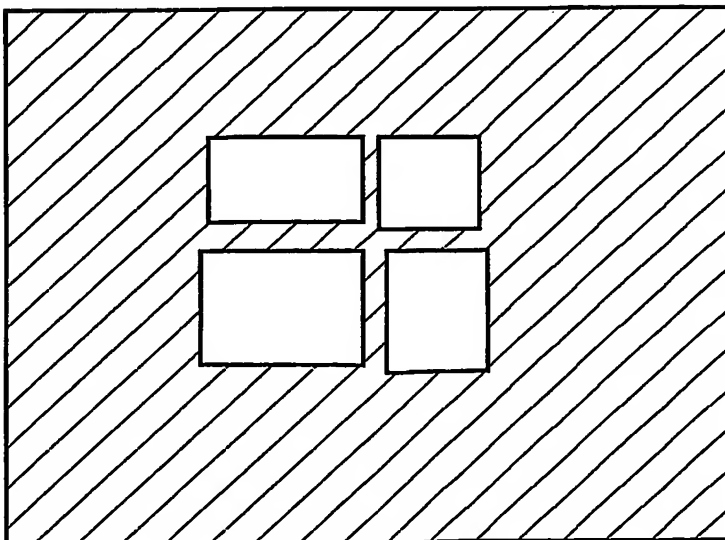
【図 3】



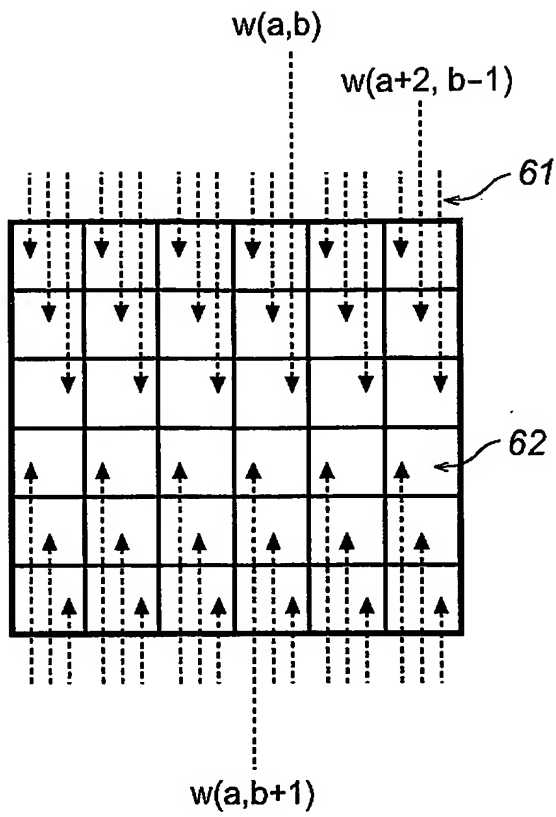
【図 4】



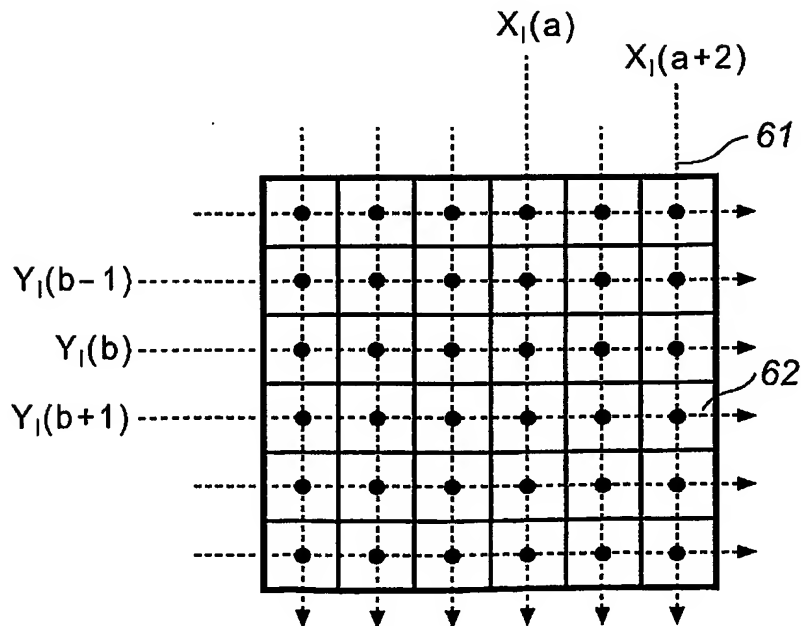
【図 5】



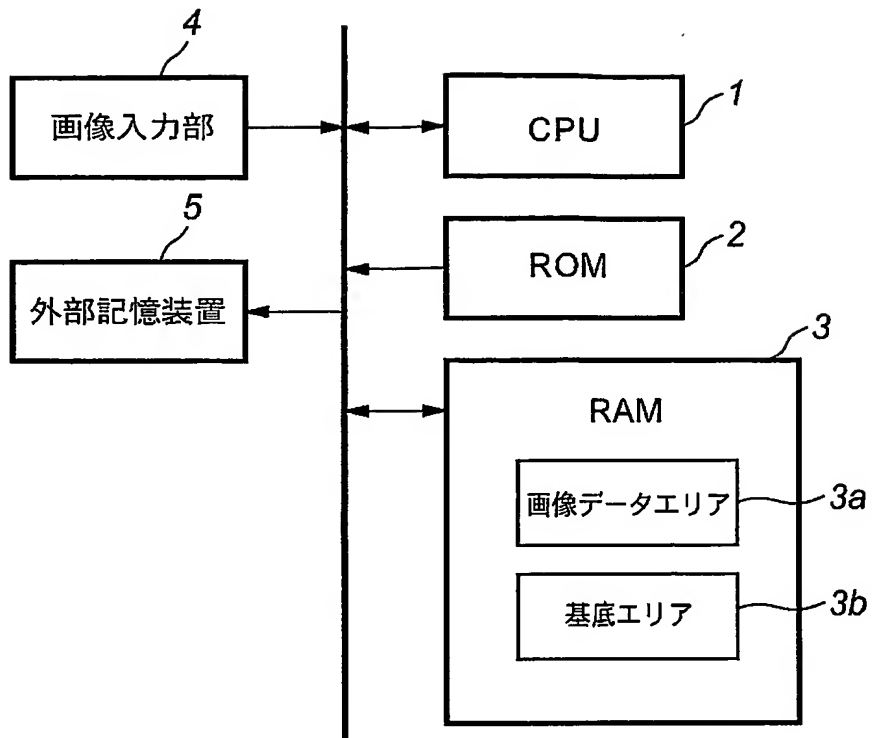
【図 6】



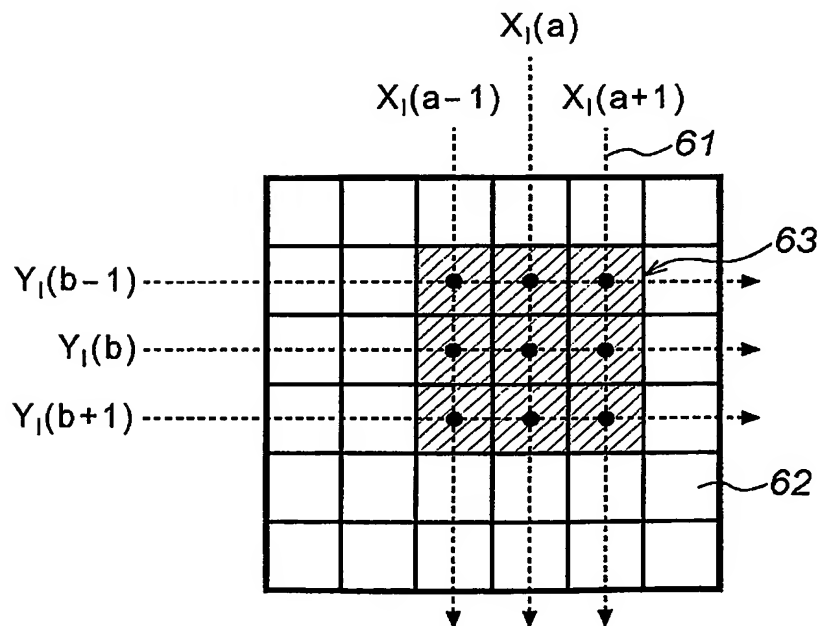
【図 7】



【図 8】



【図 9】



【図 10】

 $f(x, y)$

136	79	28	76	109	37	46	92
141	79	38	122	117	64	46	78
140	104	40	101	122	83	45	54
143	102	33	69	109	109	44	54
165	135	117	115	124	132	106	41
167	163	162	163	149	144	134	53
173	170	169	172	171	169	153	93
174	174	176	174	169	155	147	127

【図 11】

$X_l(x)_{(l=0)}$	0.1	0	0	0	0	0	0	0
$Y_l(y)_{(l=0)}$	0	0	0	0	0	0	0	0

【図 12】

 $\tilde{f}_l(x, y)_{(l=0)} = X_l(x)_{(l=0)} \cdot Y_l(y)_{(l=0)}$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

【図 13】

$\Delta X_0(x)_{(l=1)}$	0	0	0	0	0	0	0
$\Delta Y_0(y)_{(l=1)}$	0.0106	0.011	0.0109	0.0112	0.0129	0.0135	0.0136

【図 14】

$X_0(x)_{(l=1)}$	0.1	0	0	0	0	0	0
$Y_0(y)_{(l=1)}$	0.0106	0.011	0.0109	0.0112	0.0129	0.0135	0.0136

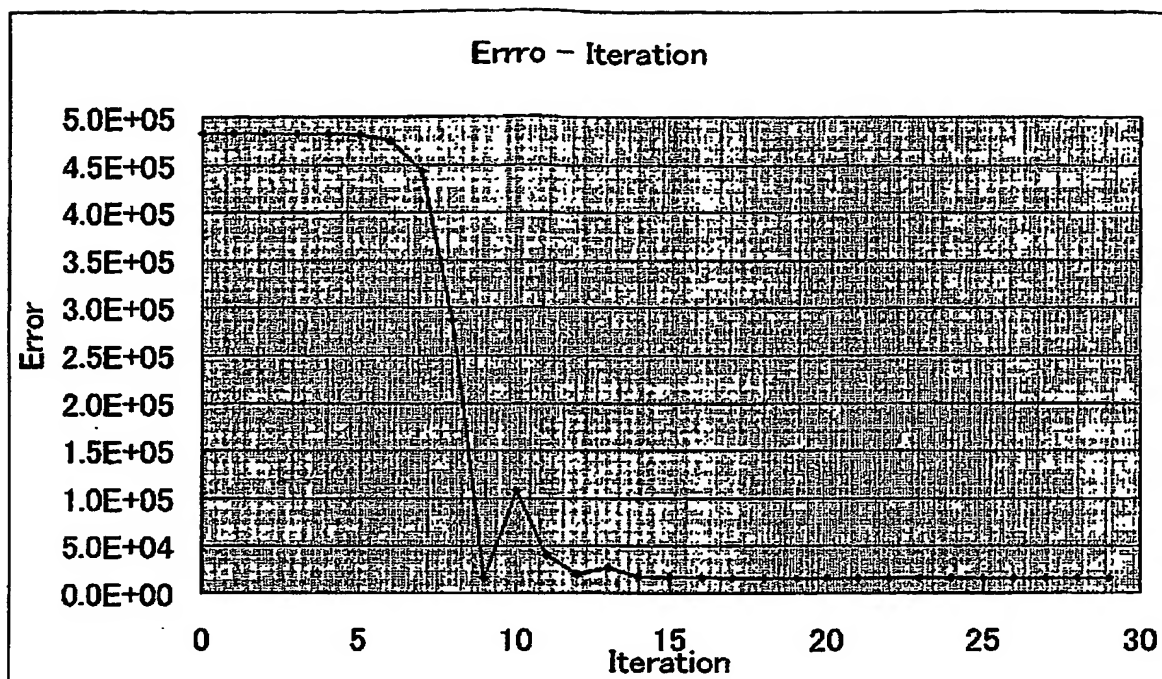
【図 15】

$\Delta X_0(x)_{(l=2)}$	0.0118	0.0098	0.0077	0.0096	0.0103	0.0087	0.0072	0.0057
$\Delta Y_0(y)_{(l=2)}$	0.0193	0.0205	0.0204	0.0206	0.0248	0.0264	0.0279	0.0281

【図 16】

$\Delta X_0(x)_{(l=9)}$	6.0845	5.228	4.3387	5.16	5.3904	4.7233	3.9623	2.9586
$\Delta Y_0(y)_{(l=9)}$	4.3238	4.9659	5.0631	4.8601	6.7611	8.1312	8.9774	9.076

【図 17】



【図 18】

$X_0(x)_{(t \rightarrow \infty)}$	9.3141	7.9686	6.6163	7.8649	8.2146	7.2	6.0413	4.5084
$Y_0(y)_{(t \rightarrow \infty)}$	10.494	12.0484	12.2836	11.7937	16.4004	19.7151	21.7626	22.0012

【図 19】

$$\tilde{f}_0(x, y)_{(t \rightarrow \infty)} = X_0(x)_{(t \rightarrow \infty)} \cdot Y_0(y)_{(t \rightarrow \infty)}$$

97.7422	83.6225	69.4315	82.5343	86.204	75.5568	63.3974	47.3111
112.22	96.0089	79.7158	94.7595	98.9728	86.7485	72.788	54.319
114.4107	97.8831	81.272	96.6093	100.9049	88.4419	74.2089	55.3794
109.8477	93.9793	78.0307	92.7563	96.8805	84.9146	71.2493	53.1707
152.755	130.6882	108.51	128.9875	134.7227	118.0829	99.0797	73.9396
183.6284	157.1017	130.441	155.0573	161.9517	141.9487	119.1048	88.8836
202.699	173.4175	143.9879	171.1607	178.7711	156.6907	131.4744	98.1145
204.9214	175.3188	145.5665	173.0372	180.7311	158.4086	132.9158	99.1902

【図 20】

$$f_1(x, y) = f(x, y) - \tilde{f}_0(x, y)_{(t \rightarrow \infty)}$$

38.2578	-4.6225	-41.4315	-6.5343	22.796	-38.5568	-17.3974	44.6889
28.78	-17.0089	-41.7158	27.2405	18.0272	-22.7485	-26.788	23.681
25.5893	6.1169	-41.272	4.3907	21.0951	-5.4419	-29.2089	-1.3794
33.1523	8.0207	-45.0307	-23.7563	12.1195	24.0854	-27.2493	0.8293
12.245	4.3118	8.49	-13.9875	-10.7227	13.9171	6.9203	-32.9396
-16.6284	5.8983	31.559	7.9427	-12.9517	2.0513	14.8952	-35.8836
-29.699	-3.4175	25.0121	0.8393	-7.7711	12.3093	21.5256	-5.1145
-30.9214	-1.3188	30.4335	0.9628	-11.7311	-3.4086	14.0842	27.8098

【図 2 1】

$\Delta X_1(x)_{(t=1)}$	0	0	0	0	0	0	0	0
$\Delta Y_1(y)_{(t=1)}$	0.003	0.0022	0.002	0.0026	0.001	-0.0013	-0.0023	-0.0024

【図 2 2】

$\Delta X_1(x)_{(t=2)}$	0.0004	-8.9691	-0.0005	-3.0473	0.0002	-9.7318	-0.0003	0.0001
$\Delta Y_1(y)_{(t=2)}$	0.0042	0.0031	0.0028	0.0037	0.0013	-0.0018	-0.0033	-0.0034

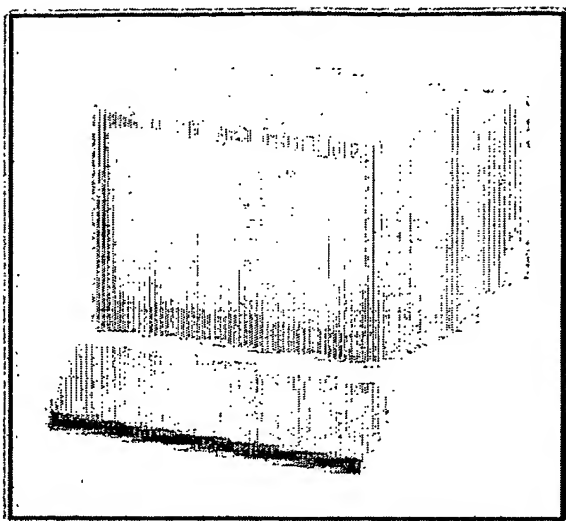
【図 2 3】

$X_0(x)_{(t \rightarrow \infty)}$	9.314059	7.968558	6.616251	7.864916	8.214631	7.200042	6.041299	4.508398
$Y_0(y)_{(t \rightarrow \infty)}$	10.49404	12.04839	12.28356	11.79374	16.4004	19.71507	21.76265	22.00115
$X_1(x)_{(t \rightarrow \infty)}$	5.855984	-0.47626	-7.69964	0.047947	3.368921	-2.28737	-4.5027	3.50781
$Y_1(y)_{(t \rightarrow \infty)}$	6.61005	5.70475	4.722223	4.527204	-1.47624	-4.08533	-3.73539	-2.88403
$X_2(x)_{(t \rightarrow \infty)}$	2.526245	1.695333	-1.70869	-2.36103	-0.26453	4.25259	-1.22741	-6.25392
$Y_2(y)_{(t \rightarrow \infty)}$	-3.30739	-2.16792	1.972279	4.676515	4.133266	1.484439	-0.85693	-4.40687
$X_3(x)_{(t \rightarrow \infty)}$	0.619973	-1.05835	0.174089	4.68368	0.360208	-2.66006	-0.4968	-3.57879
$Y_3(y)_{(t \rightarrow \infty)}$	-1.13664	3.46946	1.470049	-3.5713	0.287339	3.133781	-0.78771	-2.50734

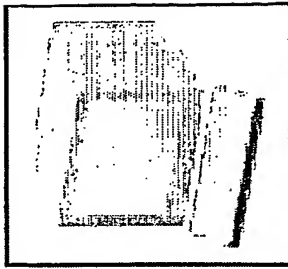
$$f_{\text{decode}}(x, y) = \sum_{n=0}^3 [X_n(x)_{(t \rightarrow \infty)} \cdot Y_n(y)_{(t \rightarrow \infty)}]$$

127.3904	76.07	23.9896	85.3368	108.9388	49.3964	38.2588	95.2501
142.3006	85.9441	40.0989	116.4015	120.0151	55.2518	48.0385	75.4716
147.9569	97.4211	41.7976	99.0642	116.8215	82.1176	49.7948	54.3484
145.9587	103.5309	34.5601	65.2056	109.6095	103.9472	46.8991	52.5858
154.7292	138.0938	112.8633	120.504	128.76	138.273	100.5108	41.8837
165.3966	158.2463	159.9045	166.0343	148.925	149.2706	134.1207	54.0541
178.171	174.5768	174.0754	169.3161	166.1308	163.6874	149.7371	93.1899
175.3439	171.8735	174.8646	171.5601	171.2779	152.9351	152.5561	125.6068

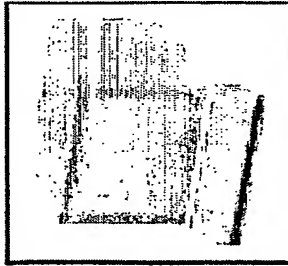
【図 24】



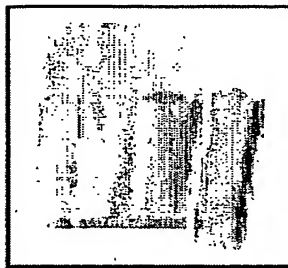
【図 25】



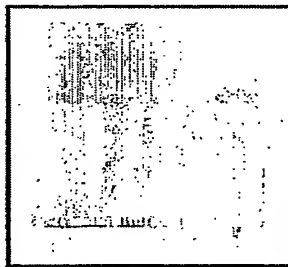
20



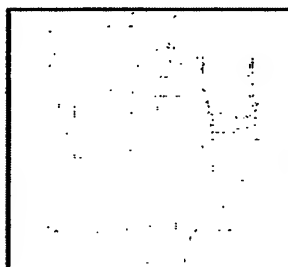
10



5



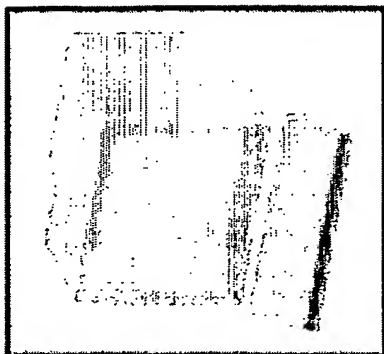
2



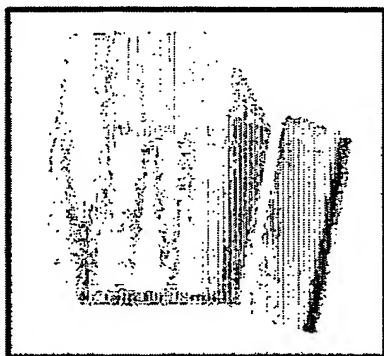
1

基底組数 =

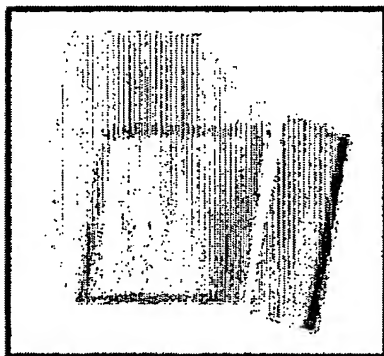
【図 26】



4ビット



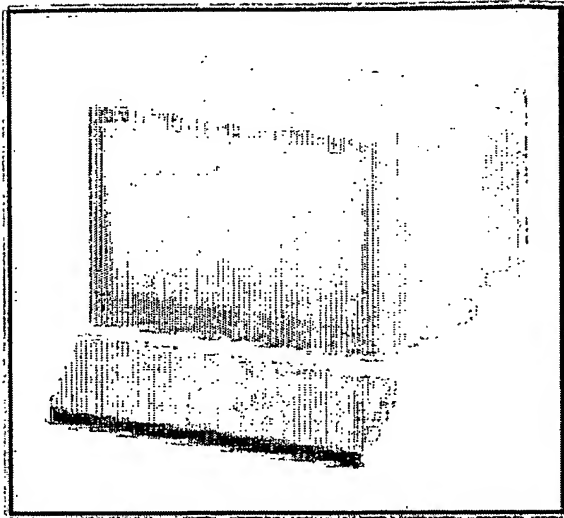
6ビット



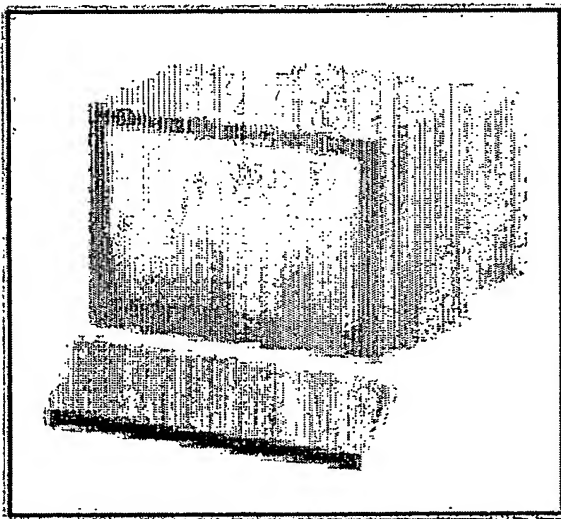
8ビット

基底要素値の
ビット数

【図 27】



【図 28】



【図 29】

$Q_x(x), Q_y(y)$	0.3	0.6	0.6	1.2	1.5	1.8	2.1	2.4
------------------	-----	-----	-----	-----	-----	-----	-----	-----

【書類名】 要約書

【要約】

【課題】 2次元画像等の多次元空間における分布データ等を、高効率に符号化することができる。また2次元のデータを1次元の基底データに分解することが可能になるため、コンボリューション演算器において問題となっていた、2次元並列化による配線の問題を解決する。

【解決手段】

符号化しようとする2次元画像データ $f(x, y)$ を入力 (S10) すると共に、その2次元画像の1次元適応化基底 $X(x)$ 、 $Y(y)$ を確保し、初期値を設定する (S11)。そして、一次元適応化基底に基づき復元画像を生成し (S12)、復元画像と入力画像との誤差 E を演算し (S13)、その誤差 E に基づいて1次元適応化基底 $X(x)$ 、 $Y(y)$ の修正量 $\Delta X(x)$ 、 $\Delta Y(y)$ を求める (S14)。そして、誤差 E が下限に達した (S16)、或いは、収束した (S17) と判断するまで、復元画像の生成と誤差を求めるを行う。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2004-239792
受付番号	50401398551
書類名	特許願
担当官	第七担当上席 0096
作成日	平成 16 年 8 月 24 日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000001007
【住所又は居所】	東京都大田区下丸子 3 丁目 30 番 2 号
【氏名又は名称】	キヤノン株式会社

【代理人】

申請人

【識別番号】	100076428
【住所又は居所】	東京都千代田区紀尾井町 3 番 6 号 秀和紀尾井町 パークビル 7 F 大塚国際特許事務所
【氏名又は名称】	大塚 康德

【選任した代理人】

【識別番号】	100112508
【住所又は居所】	東京都千代田区紀尾井町 3 番 6 号 秀和紀尾井町 パークビル 7 F 大塚国際特許事務所
【氏名又は名称】	高柳 司郎

【選任した代理人】

【識別番号】	100115071
【住所又は居所】	東京都千代田区紀尾井町 3 番 6 号 秀和紀尾井町 パークビル 7 F 大塚国際特許事務所
【氏名又は名称】	大塚 康弘

【選任した代理人】

【識別番号】	100116894
【住所又は居所】	東京都千代田区紀尾井町 3 番 6 号 秀和紀尾井町 パークビル 7 F 大塚国際特許事務所
【氏名又は名称】	木村 秀二

特願 2 0 0 4 - 2 3 9 7 9 2

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 1 0 0 7]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/001289

International filing date: 24 January 2005 (24.01.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-239792
Filing date: 19 August 2004 (19.08.2004)

Date of receipt at the International Bureau: 31 March 2005 (31.03.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record.**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.